

# **For Reference**


---

**NOT TO BE TAKEN FROM THIS ROOM**



Ex LIBRIS  
UNIVERSITATIS  
ALBERTAENSIS





Digitized by the Internet Archive  
in 2024 with funding from  
University of Alberta Library

<https://archive.org/details/Chan1973>











THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR FRANCIS K. CHAN.....  
TITLE OF THESIS DOCUMENT CLASSIFICATION AND INDEXING.....  
THROUGH USE OF FUZZY RELATIONS AND.....  
DETERMINATION OF SIGNIFICANT FEATURES.....  
DEGREE FOR WHICH THESIS WAS PRESENTED MASTER OF SCIENCE.....  
YEAR THIS DEGREE GRANTED 1973.....

Permission is hereby granted to THE UNIVERSITY OF  
ALBERTA LIBRARY to reproduce single copies of this  
thesis and to lend or sell such copies for private,  
scholarly or scientific research purposes only.

The author reserves other publication rights, and  
neither the thesis nor extensive extracts from it may  
be printed or otherwise reproduced without the author's  
written permission.





THE UNIVERSITY OF ALBERTA

DOCUMENT CLASSIFICATION AND INDEXING THROUGH USE OF FUZZY  
RELATIONS AND DETERMINATION OF SIGNIFICANT FEATURES

BY



FRANCIS K. CHAN

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH  
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE  
OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTING SCIENCE

EDMONTON, ALBERTA

SPRING, 1973





THE UNIVERSITY OF ALBERTA  
FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and  
recommend to the Faculty of Graduate Studies and Research, for  
acceptance, a thesis entitled .....  
Indexing Through Use of Fuzzy Relations and Determination .....  
of Significant Features .....  
submitted by Francis K. Chan .....  
in partial fulfilment of the requirements for the degree of  
Master of Science .....





## ABSTRACT

The proposed automatic document classification system can be roughly divided into three phases, namely,

- 1) Feature Extraction Phase,
- 2) Feature Selection and Ordering Phase,
- 3) Classification Phase.

The feature extraction phase is designed to extract the maximum possible number of features from the sample documents of the data base. The max - min composition operation in fuzzy logic is used to extract the strongest possible relations between features. To maximize the efficiency of the proposed classification system, a feature selection and ordering phase is included to determine those significant features that contribute most to the classification process. A feature selection technique based on the Karhunen - Loève expansion scheme is applied. A parametric training method is used to train the document classifier in the classification phase. Sample statistics are collected from the sample documents of each individual class. Use of discriminant functions based on statistical relations between keywords and classes forms the basis of the classification process.





## ACKNOWLEDGEMENT

It is the author's pleasure to acknowledge appreciation to Professor H. S. Heaps for his invaluable guidance and financial support at all stages of the research. Thanks are due to Dr. L. K. Schubert for suggesting the thesis project. The author also owes special thanks to Miss Shirley Chung for her constant encouragements.

April, 1973

F. Chan

University of Alberta

Edmonton, Alberta.



# TABLE OF CONTENTS

CHAPTER		PAGE
I.	INTRODUCTION .....	1
	1.1. Statement of the Problem .....	1
	1.2. Important Developments in Automatic Document Classification .....	3
	1.3. Proposed Classification System .....	7
II.	AUTOMATIC CLASSIFICATION SYSTEM .....	9
	2.1. General .....	9
	2.2. Feature Extraction Stage .....	11
	2.2.1. Human or Logical Design Technique .....	12
	2.2.2. Statistical Feature Extraction	12
	2.3. Feature Selection Stage .....	13
	2.3.1. Information Theoretic Approach	14
	2.3.2. Direct Estimation of Error Probability .....	19
	2.3.3. Feature Space Transformation .	23
	2.3.4. Stochastic Automata Approach .	24
	2.3.5. Comparisons of Different Methods for Feature Selection	27
	2.4. Pattern Classification Stage .....	29
	2.4.1. Parametric Training Method ...	31
	2.4.2. Nonparametric Training Method	36
III.	FUZZY LOGIC .....	38





CHAPTER		PAGE
3.1.	General .....	38
3.2.	Basic Definitions of Fuzzy Sets .....	39
3.3.	Algebraic Operations on Fuzzy Sets ...	43
3.4.	Fuzzy Relation .....	45
3.5.	Similarity Relation .....	47
3.6.	Feature Extraction Based on Fuzzy Relation .....	50
IV.	KARHUNEN - LOÈVE EXPANSION .....	55
4.1.	General .....	55
4.2.	The Generalized Karhunen - Loève Expansion .....	56
4.2.1.	Derivation of the Generalized Karhunen - Loève Expansion ...	57
4.3.	Optimal Properties of the Generalized Karhunen - Loève Expansion .....	59
4.3.1.	Derivation of the First Property .....	59
4.3.2.	Derivation of the Second Property .....	62
4.4.	Discrete Equivalent of the Generalized Karhunen - Loève Expansion .....	64
4.5.	Practical Application of the Generalized Karhunen - Loève Expansion	66



4.5.1.	Necessary and Sufficient Conditions for the Generalized Karhunen - Loève Expansion ...	67
4.6.	Procedure for Formulation of the Karhunen - Loève System .....	68
V.	DATA BASE .....	72
5.1.	The CACM Data Base .....	72
5.2.	Selection of Keywords .....	74
5.3.	Selection of Classes .....	76
5.4.	Statistics of the CACM Data Base .....	77
VI.	THE PROPOSED CLASSIFICATION SYSTEM .....	79
6.1.	Introduction .....	79
6.2.	Feature Preselection Phase .....	80
6.3.	Association Measures Assignment Phase	82
6.3.1.	Document Term Matrix .....	82
6.3.2.	Term Connection Matrix .....	83
6.3.3.	Term Relation Matrix .....	83
6.4.	Complete Fuzzy Relations Assignment Phase .....	85
6.5.	Feature Selection and Feature Ordering Phase .....	88
6.5.1.	Feature Vector .....	89
6.5.2.	Mean Feature Vector .....	89
6.5.3.	The Covariance Matrix .....	90
6.5.4.	The Transformation Matrix ....	90
6.5.5.	Feature Ordering .....	91





CHAPTER	PAGE
6.5.6. The Compressed Data Base .....	91
6.6. Sample Statistics Estimation Phase ...	93
6.7. Classification Phase .....	94
VII. RESULTS AND STATISTICS .....	96
7.1. Test Data .....	96
7.2. Programming Details .....	97
7.3. Experimental Results .....	99
7.4. Discussions and Suggestions .....	100
VIII. CONCLUSIONS .....	103
* * *	
REFERENCES .....	107
APPENDIX 1. ....	113
APPENDIX 2. ....	118
APPENDIX 3. ....	122
APPENDIX 4. ....	130



## LIST OF TABLES

Table	Description	Page
I.	Distributions of the Sample Documents in the 5 Classes .....	78
II.	Experimental Classification Results .....	101





# LIST OF FIGURES

Figure		Page
1.	A Pattern Recognition Machine .....	9
2.	A Typical Automatic Pattern Classification System .....	11
3.	Automaton Operating in Random Environment ...	25
4.	State Transition Diagram of $A_{r,k}$ Model .....	26
5.	A Pattern Classifier .....	31
6.	A Linear Classifier .....	37
7.	Diagram Illustrating the Union and Intersection of Two Fuzzy Sets .....	40
8.	Parallel and Series Connection of Water Pipes Simulating Union and Intersection of Two Fuzzy Sets .....	42
9.	A Network of Water Pipes Simulating $[\{\mu_1(x) \wedge \mu_2(x)\} \vee \mu_3(x)] \vee [\{\mu_4(x) \vee \mu_5(x)\} \wedge \mu_6(x)]$ ..	43
10.	Relation Matrix Having Similarity Relations .	49
11.	1 - step Fuzzy Relation Matrix of Elements $\mu_1(x_i, x_j)$ 's .....	52
12.	1 - step Fuzzy Relation Diagram for $\mu_1(x_i, x_j)$ 's	52
13.	2 - step Fuzzy Relation Matrix of Elements $\mu_2(x_i, x_j)$ 's .....	52
14.	Complete Fuzzy Relation Diagram for $\mu(x_i, x_j)$ 's	53
15.	Flowchart Showing the Procedure for Formulatio n of the Karhunen - Loève System .....	71
16.	The Flow Diagram for the Feature Preselection Phase .....	81



Figure		Page
17.	The Flow Diagram for the Association Measures Assignment Phase .....	85
18.	The 1 - step Fuzzy Relation Matrix .....	86
19.	The Flow Diagram for the Complete Fuzzy Relations Assignment Phase .....	88
20.	The Flow Diagram for the Feature Selction and Feature Ordering Phase .....	92





## CHAPTER I

### INTRODUCTION

#### 1.1. Statement of the Problem.

In the past two decades, the rate of growth of information concerned with scientific fields has increased in a manner often described as leading to an "information explosion" (1). A large number of scientific and technical papers are produced each day, and the existing manual classification systems cannot handle the resulting large amount of material. A knowledge of up-to-date information is particularly essential in the fields of science; indeed it is directly responsible for the further development of many new technologies. With the advent of high speed electronic computers, scientists began to think of using mechanical means to substitute for the intellectual task of document classification.

In Canada, the problem of the information explosion is increased by the shortage of well-trained librarians; only a few universities can produce a limited number of library science graduates every year. The only alternative at hand is to switch to automatic classification systems, thus using computers to remedy the shortage of qualified manual classifiers.

Mechanical classification has several advantages over manual systems. A large store of computer accessible data may be stored with relative ease. Unlike humans, a computer can be programmed to deal with several fields at the same



time. No human can do this with comparable efficiency.

Stability is a desirable feature in classification systems. In the conventional manual classification systems, the classification is almost always influenced by the classifier's background, attitude, and disposition. The quality of classification may therefore vary widely among classifiers. Even if the same person attempts to repeat his classification of a document at a later date, he may well produce a different resulting classification. It is obvious that the result of a manual classification is likely to be greatly biased. It is subjective and is bound to be affected by the classifier's own opinions and his current interests.

The four most common manual classification schemes, namely, the Universal Decimal Classification (UDC) (2), Library of Congress Classification (LC) (2), Dewey Decimal Classification (DC) (2), and Colon Classification (CC) (2), are notorious for their inefficiency in classifying highly specialized subjects; none of them have the required versatility to represent a complex scientific document (3). Information scientists realize that, instead of developing a more sophisticated new classification scheme, they have to tackle the problem by an entirely new approach, and using a computer may well produce the required answer.

With a computerized automatic document classification system, the above mentioned problems in manual classification may readily be tackled. The speed of a highly sophisticated modern electronic computer can also allow other desirable





features. A computerized automatic classification system offers a higher level of searching efficiency, which results in an enormous time saving. In most libraries a surprisingly large amount of time is required to classify each document by manual means. Also, such classification is very expensive since the major expense of a manual classification system is the required salaries of the specialized personnel involved.

A computerized classification also has the advantage of being dependable in the sense that, once an automatic classification system has been created, the system is there to stay and will provide the required service at all times.

#### 1.2. Important Developments in Automatic Document Classification.

H. P. Luhn first suggested a statistical approach to mechanize encoding and searching of literary information in 1957 (4). He used the fact that different combinations of words may be used to convey the same idea. Using a statistical approach, Luhn suggested a statistical analysis of a collection of documents within a particular field of interest. He applied the statistical results to set up a thesaurus-type dictionary to be used to encode and search literary information. Luhn suggested that particular combinations of words should indicate the significant concepts within a document, and that the frequency of word occurrences should provide a useful measurement of word significance (5).

H. P. Luhn furthered his research in 1958 and applied



his theory to automatic creation of literature abstracts (6). For each document some statistical information is derived from word frequencies and distributions to give a measure of relative significance of individual words and sentences. Those sentences that score the highest in significance are extracted and combined to form the abstract. The theory is based on the fact that the significant factors of a sentence can be derived solely from an analysis of its keywords. It is supposed that there is only a small probability that the writer of a technical paper would use different words to reflect the same notion, and that there is only a small probability that he would use the same word to reflect more than one different notion in the same paper.

Inspired by the work of Luhn, in 1960 M. E. Maron and J. L. Kuhns studied the statistical relationship of words within a group of documents. Their approach was based on measurement of the relative frequencies with which the words appear (7). Given a request for information, a probabilistic indexing scheme makes a statistical inference on documents in the data base, and derives a relevance number for each document. This number provides a measure of the probability that the document will satisfy the request. The result of the search is then an ordered list of the documents which satisfy the request. Documents within the list are ranked in importance according to their probable relevance.

In the same year (1960), G. Salton at Harvard University created a retrieval system called "SMART"



(SALTON'S MAGICAL AUTOMATIC RETRIEVAL TECHNIQUE) (8, 9).

Several hundred forms of analysis were employed to examine each document with a view to obtaining those words that are most suitable for representation of, and search on, the document. Some of the techniques used were statistical word association, syntactic analysis, statistical phrase recognition, and hierachical arrangement of concepts.

A statistical approach to measure the probability of the relationships between words and classification categories was again used by Maron in 1961 (10). Maron developed a formula to compute an "attribute number" which gives the probability that a document indexed by a certain combination of keywords will belong to a certain class. He then attempted to use attribute numbers to classify documents according to their subject content.

H. E. Stiles studied the probability of constructing an entirely automatic computer document retrieval system in 1961 (11). By application of certain statistical formulae, Stiles proposed to calculate the degree of association between pairs of index words in the data base in terms of their frequencies of occurrence. His system matches request terms and the terms used to index a document according to their degree of association. The documents selected are arranged in order of their relevance to the request.

L. B. Doyle also published a paper in 1962 (12) which discussed the inter-relationships between words within a document.





F. B. Baker viewed the problem of automatic document classification from an entirely new angle in 1962 (13). Baker applied the latent class analysis, first suggested by Lazarsfeld (14), to document classification. The analysis is based on a mathematical model derived on the assumption that a set of data described by statistics may be divided into small subsets, such that in each subset the probabilities of different word incidences are statistically independent.

Words chosen to describe documents are not always the most satisfactory for document retrieval. Recognizing this, in 1963 G. Salton suggested use of associative document retrieval techniques using bibliographic information (15). Salton argued that documents that exhibit similar citation sets are likely to deal with similar subject matter. The addition of bibliographic information to other standard criteria should therefore prove to be a valuable asset in automatic document classification. E. Garfield also proposed use of the notion of bibliographic links in document retrieval. By the aid of computer techniques, Garfield applied the idea of citation index in literature searching and he set up an important scientific literature retrieval service known as the Science Citation Index (16).

During 1963 and 1964, H. Borko and M. Bernick conducted a series of experiments in automatic document classification (17, 18). Techniques such as factor analysis, Bayesian classification method, and factor score method were applied.



### 1.3. Proposed Classification System.

The proposed classification system is a small computerized automatic document classification system suitable for a specialized scientific information centre with a limited amount of classification personnel. It is believed that the titles of scientific papers give a good indication of the contents of the papers, and thus keywords chosen from the titles are used to represent the documents. In order to obtain the maximum possible relationship between keywords, fuzzy logic is applied for feature extraction and to obtain indirect relationships to extend the possible relationships between the given keywords (19). The dimension of the resulting fuzzy relation matrix of keywords is usually very large and hence it is desirable to apply a dimension reduction process. By applying the Karhunen - Loève expansion in feature selection and ordering (20), only those keywords that are most useful for classification are retained. The process involves determination of the covariance matrix for distinct keywords in the data base, and calculation of its corresponding eigenvalues and eigenvectors. The reduction in dimension is achieved through a linear transformation. Since the statistics of documents for individual classes are readily available, a statistical approach using a maximum likelihood discriminant function is employed in the classification. The theory of the proposed classification system is based entirely on the relationship between subject categories and the document content as



indicated by the keyword statistics.

The general classification problem is, of course, dependent on the method of feature selection and the subsequent processing of the extracted features. This general problem is discussed in Chapter II. The use of fuzzy logic is described in Chapter III, and application of the Karhunen - Loève expansion is explained in Chapter IV. The remaining chapters are concerned with application to classification of documents of a specific test data base.





## CHAPTER II

### AUTOMATIC CLASSIFICATION SYSTEM

#### 2.1. General.

Modern information retrieval systems may make use of concepts developed for pattern recognition. Machines may be designed to perform conceptual recognition by use of a priori information. A pattern recognition machine, as represented in Fig. 1, may be defined as a device capable of sorting or classifying patterns. Inputs in the form of measured values are fed into the machine, and outputs are produced in the form of predictions. The criterion of success for such a machine is its ability to minimize the number of misrecognitions so that the resulting forecasts are in close agreement with the subsequently observed outcomes. Many theories of pattern recognition are derived from statistical decision theory which deals with classification of measurements; others result from research on the perceptron and adaptive decision networks.

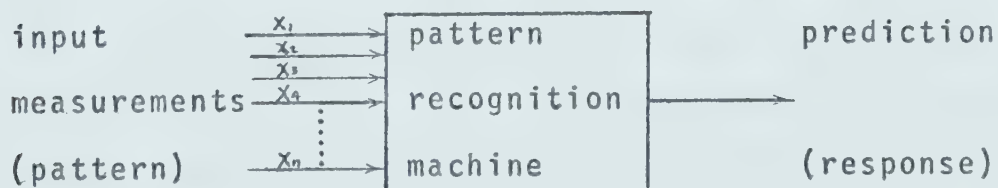


Fig. 1. A Pattern Recognition Machine.

Before proceeding further, we shall briefly review



some of the terminology commonly used in the field of pattern recognition. According to J. T. Tou, pattern recognition is defined as the categorization of input data into identifiable classes via extraction of the significant features of the data from a background of irrelevant detail (21). A pattern is a set of data to be classified. Using vector notation, a pattern may be regarded as an  $n$  - dimensional column vector whose elements represent  $n$  different properties of a pattern. Each individual property of a pattern is called a feature.

In geometric terms, the concept of pattern recognition can be expressed in terms of a partition of pattern space. A pattern may be regarded as a point in a  $d$  - dimensional Euclidean space  $E^d$ , called the pattern space. Patterns that belong to the same class correspond to an ensemble of points distributed within some recognizable region of the space. A pattern classifier attempts to group the pattern points of  $E^d$  into classes. If points of the same class cluster together then decision surfaces may divide pattern space into decision regions each of which characterizes a pattern class. The decision regions are defined by  $n$  discriminant functions  $G_i(X)$ ,  $i = 1, 2, \dots, n$ , where  $X$  is the feature vector and such that for each region  $i$  there is the inequality  $G_i(X) > G_j(X)$  for all  $i, j = 1, 2, \dots, n$  and  $i \neq j$ . The decision surface that separates region  $i$  and region  $j$  can be expressed by the equation:

$$G_i(X) - G_j(X) = 0. \quad [2.1]$$



Thus mathematically, the problem of pattern recognition may be viewed as a mapping of feature measurement vectors into proper classes.

The creation of an automatic pattern classification system may be roughly divided into three stages as shown in Fig. 2. They include,

1) Feature Extraction Stage,

2) Feature Selection Stage,

and 3) Pattern Classification Stage.

Each stage is examined in detail in the following sections.

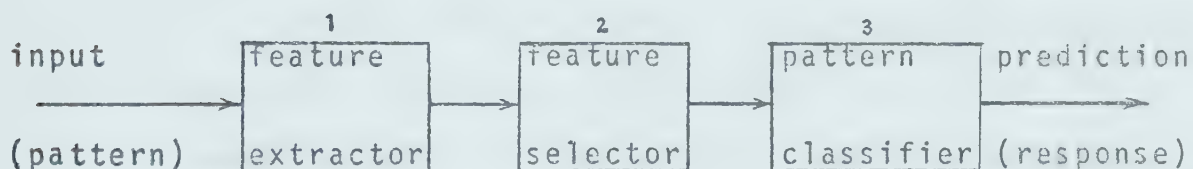


Fig. 2. A Typical Automatic Pattern Classification System.

## 2.2. Feature Extraction Stage.

Feature extraction is the first stage of an automatic classification system. Its main purpose is to extract feature measurements from input patterns, and to condition or format the input data to a form suitable for subsequent analysis of the features. The input feature measurements should cover all the information that is available about the pattern; they are expressed as numerical values whose magnitudes indicate the amount of each feature that the pattern possesses. Feature measurements are best expressed



in vector representation because this provides a geometric interpretation of the distribution of the various patterns in the pattern space. Feature extraction is important because the performance of the entire system is dependent on it. Failure to extract all available data from a pattern will result in a corresponding loss of information for processing in subsequent stages.

#### 2.2.1. Human or Logical Design Technique.

Owing to lack of general techniques for the design of feature extraction, it is necessary to take advantage of any a priori knowledge that the designer may possess regarding selection of the important features. The technique used is problem dependent, and it is directly related to the knowledge and ingenuity of the designer. For example, in applications that involve time series data, the amplitudes of specific frequencies and the correlations between frequencies are suggested as possible useful features to be extracted.

The importance of human designed feature extraction should not be under estimated. By utilizing the designer's experience it is often possible to effect enormous savings in the amount of statistical analysis required in the later stages.

#### 2.2.2. Statistical Feature Extraction.

Usually, the features extracted by human designed feature extraction from initial data are not sufficient to facilitate an efficient decision making process; hence





additional features must be extracted directly from the initial data, from other features, or from a combination of both. The techniques for extracting these additional features may be classified by two criteria - the means for choosing the subset of data or inputs from which a feature is extracted, and the means by which the function that represents the feature is chosen.

The statistical technique of discriminant analysis may readily be applied to the discovery of significant features. With certain assumptions about the distributions of the input patterns, specific function can be implemented to generate statistical features from the data.

### 2.3. Feature Selection Stage.

A pattern which is to be recognized and classified should possess a number of discriminatory properties or features. Certainly, one can use the brute-force technique of measuring all possible features and then using a large amount of time to process the measured information. However, for economic reasons, it is seldom practical to use all the available information. One has to minimize the number of features examined by the classifier, and choose only those significant features that are most helpful to the classification process. There are few theories of feature selection. It is highly problem dependent and tends to be specific to each particular application. Also, human intuition and experience may have to be involved in many instances.



Possible methods of tackling the problem are to combine the original features, to obtain a subset from the original feature set, or to discard the poor features and emphasize the important ones. The number of features to be selected depends on the desired degree of accuracy in recognition. However, insistence on higher accuracy means that more features must be observed. The interset features that represent the differences between, or among, pattern classes lead to the best characterizations of the input patterns. The intraset features common to all patterns under consideration carry no discriminatory information and may be ignored.

Correct recognition depends on the amount of discriminatory information contained in the measurements. An insufficient number of feature measurements will not give a satisfactory degree of correct recognition. On the other hand, it is usually impractical to measure a very large number of features.

### 2.3.1. Information Theoretic Approach.

Approaches based on concepts of information theory have been suggested for evaluation of the discriminatory effectiveness of features. The divergence and the average information content of pattern classes characterized by features may be used as the feature selection criteria.

The concept of divergence is closely related to the discriminatory power between two pattern classes that have Gaussian distributed feature measurements. The application



of divergence to measure the goodness of features was first proposed by T. Marill and D. M. Green in 1963 (22).

Assume that the pattern class  $W_i$ ,  $i = 1, 2, \dots, N$  and the feature vector  $X$  are distributed according to the multivariate Gaussian density function with mean feature vector  $\mu_i$  and covariance matrix  $\Sigma_i$  for pattern class  $W_i$ ,  $i = 1, 2, \dots, N$ . The conditional probability  $P(X/W_i)$  may then be defined in matrix form as (23):

$P(X/W_i)$  = the probability that a pattern in class  $W_i$   
has feature vector  $X$

$$= \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma_i|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(X-\mu_i)'\Sigma_i^{-1}(X-\mu_i)\right] \quad [2.2]$$

where  $i = 1, 2, \dots, N$ ,

$n$  is the number of elements in the feature vector,

$|\Sigma_i|$  is the determinant of  $\Sigma_i$ ,

$( )'$  denotes the transpose vector of  $( )$ ,

and  $\Sigma_i^{-1}$  is the inverse of  $\Sigma_i$ .

If we define the likelihood ratio of feature vector  $X$ :

$$\lambda(X) = \frac{P(X/W_i)}{P(X/W_j)} \quad [2.3]$$

where  $i, j = 1, 2, \dots, N$  and  $i \neq j$ .

$$\text{Let } L(X) = \log_e \lambda(X), \quad [2.4]$$

from [2.3] and [2.4], we have

$$L(X) = \log_e P(X/W_i) - \log_e P(X/W_j). \quad [2.5]$$

Substituting [2.2] into [2.5] and supposing that

$\Sigma = \Sigma_i = \Sigma_j$ , and  $\Sigma$  is symmetrical, then





$$\begin{aligned}
L(X) &= \left\{ -\frac{1}{2}(X-\mu_i)' \Sigma^{-1}(X-\mu_i) \right\} - \left\{ -\frac{1}{2}(X-\mu_j)' \Sigma^{-1}(X-\mu_j) \right\} \\
&= \left\{ -\frac{1}{2}[X' \Sigma^{-1} X - 2X' \Sigma^{-1} \mu_i + \mu_i' \Sigma^{-1} \mu_i] \right\} - \\
&\quad \left\{ -\frac{1}{2}[X' \Sigma^{-1} X - 2X' \Sigma^{-1} \mu_j + \mu_j' \Sigma^{-1} \mu_j] \right\} \\
&= -\frac{1}{2} X' \Sigma^{-1} X + X' \Sigma^{-1} \mu_i - \frac{1}{2} \mu_i' \Sigma^{-1} \mu_i \\
&\quad + \frac{1}{2} X' \Sigma^{-1} X - X' \Sigma^{-1} \mu_j + \frac{1}{2} \mu_j' \Sigma^{-1} \mu_j \\
&= X' \Sigma^{-1} (\mu_i - \mu_j) - \frac{1}{2} \Sigma^{-1} (\mu_i' \mu_i - \mu_j' \mu_j) \\
&= X' \Sigma^{-1} (\mu_i - \mu_j) - \frac{1}{2} \Sigma^{-1} (\mu_i' + \mu_j') (\mu_i - \mu_j) \\
&= X' \Sigma^{-1} (\mu_i - \mu_j) - \frac{1}{2} (\mu_i + \mu_j)' \Sigma^{-1} (\mu_i - \mu_j).
\end{aligned}$$

[2.6]

Assume that the feature vector  $X$  is from class  $W_i$ , then the expected value of  $L(X)$  is:

$$\begin{aligned}
E[L(X)/W_i] &= \mu_i' \Sigma^{-1} (\mu_i - \mu_j) - \frac{1}{2} (\mu_i + \mu_j)' \Sigma^{-1} (\mu_i - \mu_j) \\
&= [\mu_i' - \frac{1}{2} (\mu_i' + \mu_j')] \Sigma^{-1} (\mu_i - \mu_j) \\
&= [\mu_i' - \frac{1}{2} (\mu_i' + \mu_j')] \Sigma^{-1} (\mu_i - \mu_j) \\
&= (\mu_i' - \frac{1}{2} \mu_i' - \frac{1}{2} \mu_j') \Sigma^{-1} (\mu_i - \mu_j) \\
&= (\frac{1}{2} \mu_i' - \frac{1}{2} \mu_j') \Sigma^{-1} (\mu_i - \mu_j) \\
&= \frac{1}{2} (\mu_i' - \mu_j') \Sigma^{-1} (\mu_i - \mu_j) \\
&= \frac{1}{2} (\mu_i - \mu_j)' \Sigma^{-1} (\mu_i - \mu_j)
\end{aligned}$$

[2.7]

whereas, if the feature vector  $X$  is from class  $W_j$ , the corresponding value of the expected value of  $L(X)$  is:

$$E[L(X)/W_j] = -\frac{1}{2} (\mu_i - \mu_j)' \Sigma^{-1} (\mu_i - \mu_j). \quad [2.8]$$

The divergence between  $W_i$  and  $W_j$  is defined as:

$$J(W_i, W_j) = E[L(X)/W_i] - E[L(X)/W_j]. \quad [2.9]$$



Substituting [2.7] and [2.8] into [2.9]

$$\begin{aligned} J(W_i, W_j) &= \frac{1}{2}(\mu_i - \mu_j)' \Sigma^{-1}(\mu_i - \mu_j) - \left[ -\frac{1}{2}(\mu_i - \mu_j)' \Sigma^{-1}(\mu_i - \mu_j) \right] \\ &= (\mu_i - \mu_j)' \Sigma^{-1}(\mu_i - \mu_j). \end{aligned} \quad [2.10]$$

Note that if  $\Sigma = I$  (the identity matrix) in [2.10] then  $J(W_i, W_j)$  represents the squared distance between  $\mu_i$  and  $\mu_j$ .

If Bayes' decision rule is used for classifier, then for

$$P(W_i) = P(W_j) = \frac{1}{2} \text{ (only two classes),}$$

$X \in W_i$  ( $X$  belongs to class  $W_i$ ) if  $\lambda(X) \geq 1$  or  $L(X) \geq 0$ ,

and  $X \in W_j$  ( $X$  belongs to class  $W_j$ ) if  $\lambda(X) < 1$  or  $L(X) < 0$ .

The probability of misrecognition is

$$\begin{aligned} e(W_i, W_j) &= P(W_j) \cdot P[L(X) \geq 0/W_j] + P(W_i) \cdot P[L(X) < 0/W_i] \\ &= \frac{1}{2}P[L(X) \geq 0/W_j] + \frac{1}{2}P[L(X) < 0/W_i]. \end{aligned} \quad [2.11]$$

From [2.6], [2.7] and [2.9], it may be concluded that  $P[L(X)/W_i]$  is a Gaussian density function with mean  $\frac{1}{2}J(W_i, W_j)$  and variance  $J(W_i, W_j)$ . Similarly,  $P[L(X)/W_j]$  is also a Gaussian density function with mean  $-\frac{1}{2}J(W_i, W_j)$  and variance  $J(W_i, W_j)$ .

Using [2.11], it can be shown that (24):

$$\begin{aligned} e(W_i, W_j) &= \frac{1}{2} \int_0^\infty [2\eta J(W_i, W_j)]^{-\frac{1}{2}} \exp\left[-\frac{1}{2}\left\{\xi + \frac{1}{2}J(W_i, W_j)\right\}/J(W_i, W_j)\right] d\xi \\ &\quad + \frac{1}{2} \int_{-\infty}^0 [2\eta J(W_i, W_j)]^{-\frac{1}{2}} \exp\left[-\frac{1}{2}\left\{\xi - \frac{1}{2}J(W_i, W_j)\right\}/J(W_i, W_j)\right] d\xi. \end{aligned} \quad [2.12]$$

$$\text{Let } y = \frac{\xi \pm \frac{1}{2}J(W_i, W_j)}{\sqrt{J(W_i, W_j)}} \quad [2.13]$$

$$\text{then } e(W_i, W_j) = \int_{\frac{1}{2}\sqrt{J(W_i, W_j)}}^{\infty} \frac{1}{\sqrt{2\eta}} \exp\left[-\frac{y^2}{2}\right] dy. \quad [2.14]$$

It may be noted that the error in classification,



$e(W_i, W_j)$ , is a monotonically decreasing function of  $J(W_i, W_j)$ . Therefore, the features selected according to the magnitude of  $J(W_i, W_j)$  will provide a corresponding discriminatory power between  $W_i$  and  $W_j$ .

Assuming that features are mutually independent in their effect on the decision, P. M. Lewis in 1962 (25) proposed that a single number statistical function  $G_i$ , such as entropy or average information, may be used to measure the goodness of feature characterization. The value of the function  $G_i$  is obtained by evaluating the feature  $C_i$  over a large sample of patterns to be recognized. Statistics that are desirable for  $G_j$  are summarized as follows:

- 1) If  $G_i > G_j$ ; then  $C_i$  should give a larger percentage of recognition than  $C_j$ .
- 2) If  $G_i > G_j$ ; then  $C_i + c$  should have a larger percentage recognition than  $C_j + c$ , where  $c$  is a constant.
- 3) Let  $C_s$  be any set of features and let  $G_s$  be the sum of the values of the  $G_i$ 's for the features in  $C_s$ . Let  $P_s$  be the percentage of correct recognition when using the feature set  $C_s$ . Then a relation  $P_s = A \cdot G_s + B$  should be true where  $A$  and  $B$  are constants. In other words, the percentage of recognition should be a linear function of the sum of the  $G_i$  values.

There is not a single number statistic that can satisfy all these desirable features. However, Lewis has proposed



a single number statistic that can meet most of the requirements. The function is as follows:

$$G_j = \sum_{i=1}^N \sum_{k=1}^{N_j} P[W_i, C_j(K)] \log \frac{P[C_j(K)/W_i]}{P[C_j(K)]}. \quad [2.15]$$

It may be noted that if  $P[C_j(K)/W_i]/P[C_j(K)]$  approaches unity, that is, if each feature by itself is not very efficient in the recognition, then  $G_j$  may be approximated by the first term in its power series expansion:

$$G_j = \sum_{i=1}^N \sum_{k=1}^{N_j} P[W_i, C_j(K)] \left\{ \frac{P[C_j(K)/W_i]}{P[C_j(K)]} - 1 \right\}. \quad [2.16]$$

The selection of features by use of a single number statistic has proved to be quite effective in Lewis's experiments. However, two restrictions have to be observed in using this goodness measure; the features selected have to be those supplied by the designer, and these features have to be statistically independent.

### 2.3.2. Direct Estimation of Error Probability.

A knowledge of the feature distribution is not always available in most pattern recognition problems. Although the probability density structure of the feature measurements can be approximated, it is still rather difficult to obtain an exact analytical measure of feature effectiveness which directly reflects the recognition accuracy. Based on Parzen and Cacoullos's results on the techniques for estimating density functions, a nonparametric method for feature





selection was proposed (26).

The method is based on direct estimation of error probabilities from a given set of pattern training samples. Let  $R(Z; h)$  denote the rectangular parallelepiped in a  $p$  - dimensional measurement space  $\Omega_y$  centred at  $Z$ ; it is defined by the relation:

$$R(Z; h) = \{Y: z_i - h_i \leq y_i \leq z_i + h_i, \\ i = 1, 2, \dots, p; p \leq N\} \quad [2.17]$$

where  $h_1, h_2, \dots, h_p$  are positive constants.

Let  $\hat{P}(Z)$  be denoted as the estimated density function, then

$$\hat{P}(Z) = \frac{1}{n \prod_{i=1}^p (2h_i)} \{\text{number of samples falling in } R(Z; h)\} \\ = \frac{1}{n \prod_{i=1}^p h_i} \sum_{j=1}^n K\left(\frac{z_1 - x_{j1}}{h_1}, \dots, \frac{z_p - x_{jp}}{h_p}\right) \quad [2.18]$$

where the weighting function  $K(Y)$  is defined by

$$K(Y) = \begin{cases} 2^{-p} & \text{if } |y_i| \leq 1 \\ 0 & \text{otherwise} \end{cases} \quad \text{for each } i = 1, 2, \dots, p. \quad [2.19]$$

The problem of studying the estimates in [2.18] is to choose suitable  $h_i$  and  $K(Y)$  for a given number of training samples  $n$ , and to prove the consistency of the estimates when  $n \rightarrow \infty$ .

Without loss of generality, let  $U_i(y)$  be the weighting



function defined by:

$$U_i(y) = \begin{cases} 2^{-1} & \text{if } |y| < 1 \\ 0 & \text{if } |y| \geq 1 \end{cases} \quad [2.20]$$

and consider the case when the rectangular parallelepiped of [2.17] is to be a hypercube centred at  $Z$ , so that  $h_1 = h_2 \dots \dots = h_p = \sigma(n)$ . The problem is thus reduced to finding the estimate of the form:

$$\begin{aligned} \hat{P}(Z) &= \frac{1}{n \prod_{i=1}^p \sigma(n)} \sum_{j=1}^n \left[ \prod_{i=1}^p U_i \left( \frac{Z_i - X_{ji}}{\sigma(n)} \right) \right] \\ &= \frac{1}{n \sigma^p(n)} \sum_{j=1}^n \left[ \prod_{i=1}^p U_i \left( \frac{Z_i - X_{ji}}{\sigma(n)} \right) \right]. \end{aligned} \quad [2.21]$$

Essentially, [2.21] is used to estimate the density function by measuring all the distances between an activated point  $Z$  and all the sample points along each coordinate bases in the  $p$  - dimensional continuous measurement space.

Now suppose that for pattern class  $W_j$  the feature vectors  $X_j$ 's are characterized by a fixed, but unknown, probability distribution over  $S$  discrete values of the measurement space where

$$S = \prod_{i=1}^N S_i \quad [2.22]$$

and  $S_i$ ,  $i = 1, 2, \dots, N$ , is the number of discrete levels of the feature coordinate  $x_i$  which is a function of  $\sigma(n)$ .

Applying the idea in [2.22] into [2.21], and expressing it in vector form, allows the estimated density function of [2.21]



to be expressed in discrete form as:

$$\hat{p}(Z) = \frac{1}{n} \sum_{j=1}^n U\left(\frac{[(Z-X_j)'(Z-X_j)]^{\frac{1}{2}}}{\sigma(n)}\right) \quad [2.23]$$

$$\text{where } U(v) = \begin{cases} 1 & \text{if } v < 1 \\ 0 & \text{if } v \geq 1 \end{cases} \quad [2.24]$$

$$\text{and } v = \frac{[(Z-X_i)'(Z-X_i)]^{\frac{1}{2}}}{\sigma(n)}.$$

It may be noted that the parameter  $\sigma(n)$  is assumed to be a given constant and there is little freedom to control it. For a given  $\sigma(n)$  we can only increase the number  $n$  of training samples so that the estimate in [2.23] is asymptotically unbiased. Therefore, for a large sample problem, the conditional probability at the activated point  $Z$  for a given class  $W_j$  becomes

$$P(Z/W_j) = \frac{C_j(Z)}{n_j} \quad [2.25]$$

$$\text{where } C(Z) = \sum_{i=1}^n U\left(\frac{[(Z-X_i)'(Z-X_i)]^{\frac{1}{2}}}{\sigma(n)}\right). \quad [2.26]$$

If the maximum likelihood discriminant rule is used for classification, the feature selection criterion is based on the direct estimate of the minimized probability of misclassification estimated from  $n$  training samples as:



$$\bar{P}_n(\epsilon) = \sum_{i=1}^S \left\{ \sum_{j=1}^m P(Z_i/W_j) P(W_j) - \max_j [P(Z_i/W_j) P(W_j)] \right\} \quad [2.27]$$

where  $p(Z_i/W_j)$  is the conditional probability for a given class  $W_j$  estimated by [2.23] with  $n_j$  training samples from class  $W_j$ ,  $m$  is the number of pattern classes, and  $S$  is the total number of discrete points in the  $N$  - dimensional feature space.

A direct estimate of [2.27] is thus used as a feature selection criterion. The feature set of  $\alpha_k$  is considered more effective than the feature set  $\alpha_L$  if  $\bar{P}_n(\epsilon/\alpha_k) < \bar{P}_n(\epsilon/\alpha_L)$ .

### 2.3.3. Feature Space Transformation.

A linear feature space transformation technique for the feature selection problem will now be described. S. Watanabe introduced a feature space compression technique based on the Karhunen - Loève (K - L) expansion (27). K. S. Fu and Y. T. Chien generalized the transformation technique for extracting the effective features (28).

Let  $X^{(i)} = \{X_1^{(i)}, X_2^{(i)}, \dots, X_N^{(i)}\}$  be a random variable from a class  $W_i$  with zero mean. The K - L expansion allows the expression of a random process in terms of a set of orthonormal vectors. Any component of the sample vector  $X^{(i)}$  can be expressed as a function of the orthonormal set of vector  $\{y_k; k = 1, 2, \dots, N\}$  such that

$$X_\ell^{(i)} = \sum_{k=1}^N \gamma_k^{(i)} y_{k\ell} \quad [2.28]$$

where  $y_{k\ell}$  is the  $\ell$ th component of the  $k$ th orthonormal vector  $Y$  and  $\gamma_k^{(i)}$  is a coefficient of  $y_{k\ell}$  for the class  $W_i$ .





If the generalized expansion in [2.28] exists for all  $x_{\ell}^{(i)}$  where  $\ell = 1, 2, \dots, N$ , and the coefficients satisfy the conditions:

$$\sum_{i=1}^m P_i E[\gamma_k^{(i)} (\gamma_{\ell}^{(i)})'] = \begin{cases} \sigma_k^2 & \text{if } k = \ell, \\ 0 & \text{if } k \neq \ell, \end{cases} \quad [2.29]$$

then the component of the ensemble covariance function  $K = \|C_{k\ell}\|$  becomes:

$$\begin{aligned} C_{k\ell} &= \sum_{i=1}^m P_i E[X_k^{(i)} X_{\ell}^{(i)}] \\ &= \sum_{k=1}^N \sigma_k^2 Y_k Y_k'. \end{aligned} \quad [2.30]$$

The expression in [2.28] whose coordinates  $\{Y_k; k = 1, 2, \dots, N\}$  are determined by [2.30] through the covariance function  $K = \|C_{k\ell}\|$ , will be called the generalized K - L expansion. Here  $\sigma_k^2$ ,  $k = 1, 2, \dots, N$ , are the eigenvalues of the covariance function K and  $Y_k$ ;  $k = 1, 2, \dots, N$ , are the eigenvectors.

The orthonormal coordinate system produced by the K - L expansion minimizes the entropy function over the variances of the coordinate coefficients and ensures that the discriminatory information over the ensemble of the probability space is concentrated in a few coordinates by use of a linear transformation.

#### 2.3.4. Stochastic Automata Approach.

Automata with learning behavior in random environments can be used for feature selection if the automata are defined



in terms of feature subsets, and the environments are characterized by training samples and a certain decision rule.

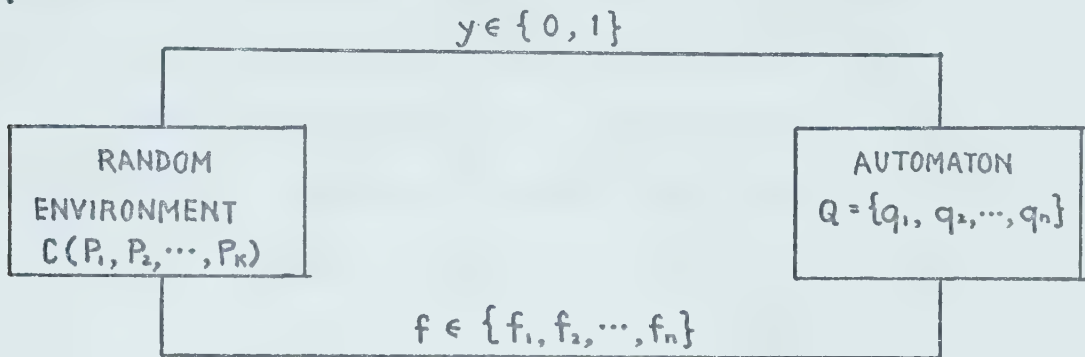


Fig. 3. Automaton Operating in Random Environment.

Fig. 3 shows the interaction between an automaton and a random environment  $C$  which is characterized by  $C(P_1, P_2, \dots, P_k)$  where  $0 \leq P_i \leq 1$  for  $i = 1, 2, \dots, k$ . The input  $y$  of the automaton can only assume two values:

$$y = \begin{cases} 0 & \text{(no penalty)} \\ 1 & \text{(penalty)}. \end{cases} \quad [2.31]$$

The output  $f$  of the automaton is its response, and  $Q$  is its internal state. In an experiment, if the automaton takes action  $f_i$ ,  $i = 1, 2, \dots, k$ , then the next input  $y$  to the machine becomes:

$$y = \begin{cases} 1 & \text{with probability } P_i \\ 0 & \text{with probability } 1 - P_i. \end{cases} \quad [2.32]$$

If  $P_i$ 's are unknown but fixed constants, it is



desirable to design an automaton with minimum expected penalty. In 1969 K. S. Fu and T. J. Li proposed a learning automaton  $A_{r,k}$  which can achieve this purpose (29). The proposed model, as shown in Fig. 4, has  $k$  actions  $f_1, f_2, \dots, f_k$  each corresponding to  $r$  states (hence the memory capacity is  $r$ ). Starting from the first state, every  $r$  consecutive states correspond to one particular action  $f_i$ .

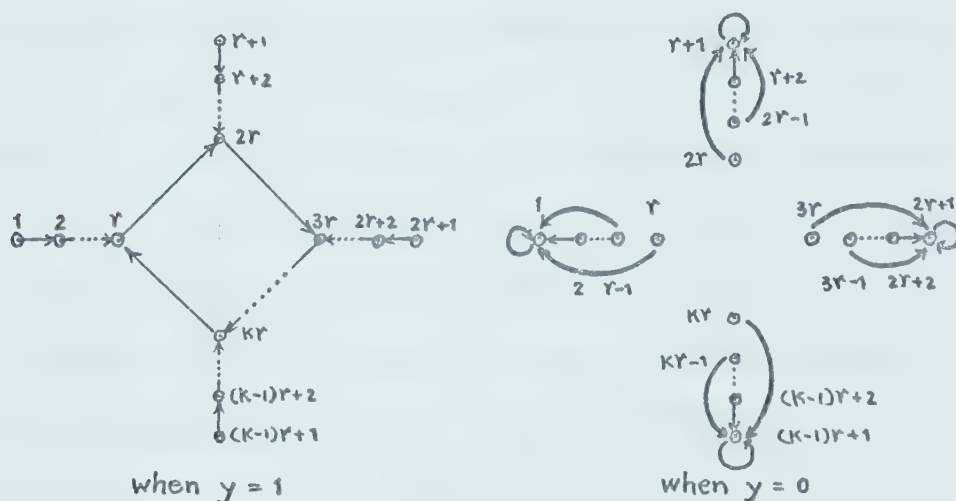


Fig. 4. State Transition Diagram of  $A_{r,k}$  Model.

It can be shown that the expected penalty for model  $A_{r,k}$  operating in the random environment  $C = C(P_1, P_2, \dots, P_k)$  is (29)

$$M(A_{r,k}; C) = \frac{\sum_{i=1}^k \frac{1}{P_i^{r-1}}}{\sum_{j=1}^k \frac{1}{P_j^r}}. \quad [2.33]$$

It may be noted that  $M(A_{r,k}; C)$  is monotonically decreasing with respect to the memory capacity  $r$ . The expected behavior is preserved even with  $r = 1$ , that is, the strategy due to model  $A_{r,k}$  is always better than a pure



random strategy. Furthermore, asymptotic optimality is achieved in the sense that  $\lim_{r \rightarrow \infty} M(A_{r,k}; C) = \min(P_1, P_2, \dots, P_K)$ . In other words, during the experiment the probability of applying the best action tends to unity as the memory capacity  $r$  increases indefinitely.

Owing to the learning property of  $A_{r,k}$ , the model can be employed as a feature selection scheme. Each possible subset of feature measurements selected is considered to be an action  $f_i$  taken by the automaton. The input of the automaton is 1, or 0, depending on whether an incorrect, or correct, classification is made on the basis of the selected feature subset. If the state transition rule of the  $A_{r,k}$  model is used, then the optimal action, or the feature subset corresponding to the minimum probability of misrecognition, will be selected most frequently. Thus, the relative effectiveness of the feature subsets can be measured by their relative frequency of occurrence.

#### 2.3.5. Comparisons of Different Methods for Feature Selection.

When the features from each class are distributed according to Gaussian probability density functions with unequal covariance matrices, we can use the minimax linear discriminant functions to derive a separability measure for the problem of feature selection in parametric multiclass pattern recognition. It appears that in general, when the covariance matrices are unequal and the overall misrecognition is used as a measure of feature effectiveness,





the separability measure based on minimax linear discriminants could be useful criterion of feature selection in multiclass pattern classification.

Direct estimation of error probability is a nonparametric feature selection technique. Since no assumptions are made for the probability structure of the feature distributions and the independence of measurements, the proposed nonparametric technique is more effective than the parametric feature selection technique when little a priori knowledge of feature distribution is available for each class. Moreover, when the number of training samples is small, the nonparametric method based on density approximation can produce comparatively good results. The computation time required for the nonparametric method of feature selection is, in general, more than that of the parametric method.

The feature selection technique based on the generalized Karhunen - Loève expansion is superior due to the fact that the transformation procedure is less sensitive with respect to the probability structure of each pattern class, which, in practice, may only be estimated from a limited number of training samples. However, the optimality of the technique is defined over the ensemble of classes, and it makes no explicit provision for the discriminatory analysis between classes. For all  $P \leq N$ , where  $N$  is the total number of features extracted, the transformed  $P$  - dimensional feature space is less effective than the same dimensional feature subspace selected by the parametric



feature selection technique based on linear discriminant functions and the nonparametric feature selection technique based on direct error estimations. When the class distributions are unknown, the feature space transformation technique is superior to the parametric feature selection technique in multiclass pattern recognition. Considering the computation time required, the feature space transformation technique has some advantages over the proposed nonparametric feature selection technique based on direct error estimations. However, to obtain an overall optimum performance of the pattern recognition system, the nonparametric method of feature selection technique is preferred.

The main advantage of the stochastic automata model for feature selection is its simplicity in implementation. Furthermore, the approach is also nonparametric in nature and able to match the decision rule used by the classifier. The basic concept is to reformulate the feature selection problem as a decision problem with adaptive behavior.

#### 2.4. Pattern Classification Stage.

Pattern classification involves the determination of an optimal decision procedure in the identification and classification process. A classifier can be regarded as a black box with  $n$  input lines and a single output line as shown in Fig. 5. The feature measurements of a pattern constitute the  $n$  inputs and the prediction constitutes the output or response. The machine attempts to decide to which pattern



class the observed data belong. Since the pattern classes can be represented by  $n$  disjoint regions in the feature space, the classifier generates decision boundaries in the form of discriminant functions to separate the  $n$  pattern classes. The discriminant functions are scalar and single valued functions estimated from the observed sample feature measurement vectors such that if  $G_i(X)$  has its largest value for a pattern  $X$  then  $X$  is assigned to class  $W_i$ . The percentage of correct recognitions depends on the effective utilization of the available discriminatory information. Based on the correlation between them, the patterns that possess the same mathematical or statistical features are grouped in the same class. The decision functions can be generated in a variety of ways. When a complete a priori knowledge about the patterns to be recognized is available, the decision functions can be determined with precision. When only qualitative knowledge about the patterns is available, reasonable guesses of the forms of the decision functions can be made. In this instance the decision boundaries may be far from correct and it may be necessary to design the machine to achieve the desirable performance through a sequence of adjustments. Machines for recognizing such patterns are best designed by use of a training procedure. Two training methods are proposed for adjusting the discriminant functions, namely,

- 1) Parametric Training Method,
- and 2) Nonparametric Training Method.



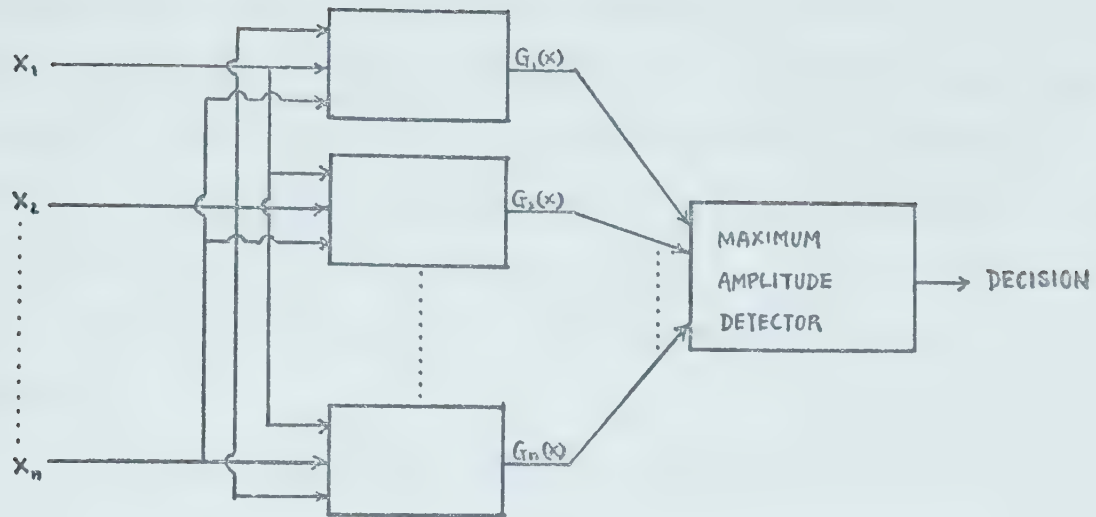


Fig. 5. A Pattern Classifier.

#### 2.4.1. Parametric Training Method. (30)

Pattern classification can be treated as a statistical decision problem. Assuming that the feature measurements of the patterns are normally distributed random variables, the joint probability density of  $n$  components of the feature measurement vector is then characterized by a multivariate normal distribution. For the  $i$ th pattern class under consideration, the normal probability density function is completely specified by the mean vector  $\mu_i$  and covariance matrix  $\Sigma_i$ . Sample patterns can be used to estimate these two parameters for each class, and the formation of the discriminant functions are based on these estimated values.

It is found that the patterns in  $n$  classes are governed by  $n$  distinct probability functions  $P(X/W_i)$ ,  $i = 1, 2, \dots, n$ . The function  $P(X/W_i)$  is defined as the probability of occurrence of pattern  $X$  given that it belongs to class  $W_i$ . There is another probability  $P(W_i)$  which is the a priori probability of occurrence of class  $W_i$ . The discriminant





function for class  $W_i$  can be expressed in terms of  $P(X/W_i)$  and  $P(W_i)$ , both of which may be estimated from the pattern samples. There exists a loss function  $\lambda(W_i/W_j)$  which is defined as the loss incurred when a pattern belonging to class  $j$  is misplaced by the system into class  $i$ . When  $\lambda(W_i/W_j)$  is minimized in some sense for all  $i$  the system is said to be optimal.

The conditional average loss when  $X$  occurs is defined as

$$L_x(W_i) = \sum_{j=1}^n \lambda(W_i/W_j) \cdot P(W_j/X) \quad [2.34]$$

where  $P(W_j/X)$  is the probability that a given pattern  $X$  belongs to class  $W_j$ . The  $L_x(W_i)$  may be calculated for any specific  $X$  with all possible values of  $i$ ,  $i = 1, 2, \dots, n$  and  $i \neq j$ .

For a specific  $X$  if  $L_x(W_i)$  is minimum of all the  $L_x(W_j)$ ,  $j = 1, 2, \dots, n$ , that is, if

$$L_x(W_j) \leq L_x(W_i), \quad i = 1, 2, \dots, n; i \neq j, \quad [2.35]$$

then the system would place  $X$  into class  $W_j$ .

By the Bayes' rule, the a posteriori probability  $P(W_j/X)$  is given by:

$$P(W_j/X) = \frac{P(X/W_j) \cdot P(W_j)}{P(X)} \quad [2.36]$$

where  $P(X/W_j)$  is the likelihood of  $W_j$  with respect to  $X$ ,  
 $P(W_j)$  is the probability of occurrence of Class  $W_j$  document,



and  $P(X)$  is the probability of occurrence of pattern  $X$ .

Substituting [2.36] into [2.34] gives the conditional average loss:

$$\begin{aligned}
 L_X(W_i) &= \sum_{j=1}^n \lambda(W_i/W_j) \cdot P(W_j/X) \\
 &= \sum_{j=1}^n \lambda(W_i/W_j) \frac{P(X/W_j) \cdot P(W_j)}{P(X)} \\
 &= \frac{1}{P(X)} \sum_{j=1}^n \lambda(W_i/W_j) \cdot P(X/W_j) \cdot P(W_j). \quad [2.37]
 \end{aligned}$$

It may be noted that  $P(X)$  is independent of  $i$ , and therefore

$$L_X(W_i) \triangleq \sum_{j=1}^n \lambda(W_i/W_j) P(X/W_j) P(W_j). \quad [2.38]$$

If we assume that an error in classification is equivalent to a unit loss, we can define  $\lambda(W_i/W_j)$  to be a special loss function known as the (0 - 1) loss function or symmetrical loss function such that

$$\lambda(W_i/W_j) = 1 - \partial_{ij} \quad [2.39]$$

where  $\partial_{ij}$  is a Kronecker delta function such that

$$\partial_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j. \end{cases} \quad [2.40]$$

Substituting [2.39] into [2.38], we have

$$\begin{aligned}
 L_X(W_i) &\triangleq \sum_{j=1}^n \lambda(W_i/W_j) \cdot P(X/W_j) \cdot P(W_j) \\
 &\triangleq P(X) - P(X/W_i) \cdot P(W_i). \quad [2.41]
 \end{aligned}$$



It is obvious that we can minimize the conditional average loss by maximizing  $P(X/W_i) \cdot P(W_i)$ . Therefore, the maximum likelihood decision can be defined as

$$G_i(X) = P(X/W_i) \cdot P(W_i). \quad [2.42]$$

Equivalently,

$$G_i(X) \triangleq \log_e [P(X/W_i) \cdot P(W_i)]$$

or

$$G_i(X) \triangleq \log_e P(X/W_i) + \log_e P(W_i). \quad [2.43]$$

The function  $P(W_i)$  is readily computed by counting the occurrence of class  $W_i$  elements in the sample training patterns. The conditional probability  $P(X/W_i)$  is obtained by making the assumption that the pattern components are normally distributed random variables. This assumption is not always met; however, the results are often found to be superior to those dependent on other assumptions.

Based on normalized distance, the multivariate normal pattern is taken to have an  $n$  - variate normal probability distribution

$$P(X/W_i) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma_i|^{\frac{1}{2}}} \exp \left\{ -\frac{1}{2} (X - \mu_i)' \Sigma_i^{-1} (X - \mu_i) \right\} \quad [2.44]$$

where the pattern  $X$  is a  $n \times 1$  column vector of the form:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$



and  $\mu_i$  is the mean feature vector estimated by

$$\hat{\mu}_i = \frac{1}{N} \sum_{x \in X_i} x_i.$$

$$\Sigma_i = \begin{bmatrix} \sigma_{11} & \dots & \dots & \sigma_{1n} \\ \vdots & \sigma_{22} & & \vdots \\ \vdots & & \ddots & \vdots \\ \sigma_{n1} & \dots & \dots & \sigma_{nn} \end{bmatrix} \text{ is a symmetric positive definite}$$

covariance matrix where

$$\hat{\Sigma}_i = \frac{1}{N} \sum_{x \in X_i} [(x_i - \hat{\mu}_i)'(x_i - \hat{\mu}_i)],$$

$\Sigma_i^{-1}$  is the inverse of  $\Sigma_i$ ,

$|\Sigma_i|$  is the determinant of  $\Sigma_i$ ,

and  $N$  is the number of sample feature vectors.

Substituting [2.44] into the expression for optimal classifier for normal pattern, we have

$$\begin{aligned} G_i(X) &= \log_e P(X/W_i) + \log_e P(W_i) \\ &= \log_e P(W_i) - \frac{n}{2} \log_e 2\pi - \frac{1}{2} \log_e |\Sigma_i| \\ &\quad - \frac{1}{2} [(X - \mu_i)' \Sigma_i^{-1} (X - \mu_i)]. \end{aligned} \quad [2.45]$$

It may be noted that  $\log_e P(W_i) - \frac{1}{2} \log_e |\Sigma_i|$  does not depend on the particular pattern being classified; it may be treated as a constant and denoted by  $b_i$ . Redefining  $G_i(X)$  to exclude the common term  $(-\frac{n}{2} \log_e 2\pi)$  for all  $i$ , we have

$$G_i(X) \triangleq b_i - \frac{1}{2} [(X - \mu_i)' \Sigma_i^{-1} (X - \mu_i)] \quad [2.46]$$

where  $b_i = \log_e P(W_i) - \frac{1}{2} \log_e |\Sigma_i|$ .





#### 2.4.2. Nonparametric Training Method. (30)

When no assumptions can be made about the characterizing parameters of individual class, a nonparametric training method may be used in pattern recognition. The method is a distributive free adaptive procedure designed to find the most appropriate weight vectors for the discriminant functions. When applying a nonparametric training method, functional forms have to be assumed for the discriminant functions. Three possible forms for the discriminant functions are:

- 1) Linear,
- 2) Quadratic,
- and 3) Piecewise Linear.

These functions all contain unspecified coefficients which constitute the weight vectors. For example, in the linear case, the linear discriminant function is defined as an equation having the form  $G_i(X) = W_1 X_1 + W_2 X_2 + \dots + W_n X_n + W_{n+1}$  with  $(W_1, W_2, \dots, W_n, W_{n+1})$  being the weight vector. A typical linear classifier using linear discriminant functions is shown in Fig. 6. In practice, the proper values for the weights are unknown; initial estimations have to <sup>be</sup> made and the machine is designed to adjust the weight vectors of the discriminant functions until the machine performs adequately on the training sets.

Nonparametric training is applicable to a wide variety of distributions; we can control the complexity of the classifier by prior specification. However an optimum performance on training set does not guarantee a similar



performance on other data. The performance of the classifier is directly proportional to the number of sample patterns being observed. Therefore, as the number of samples approaches infinity, the classifier should give perfect recognition.

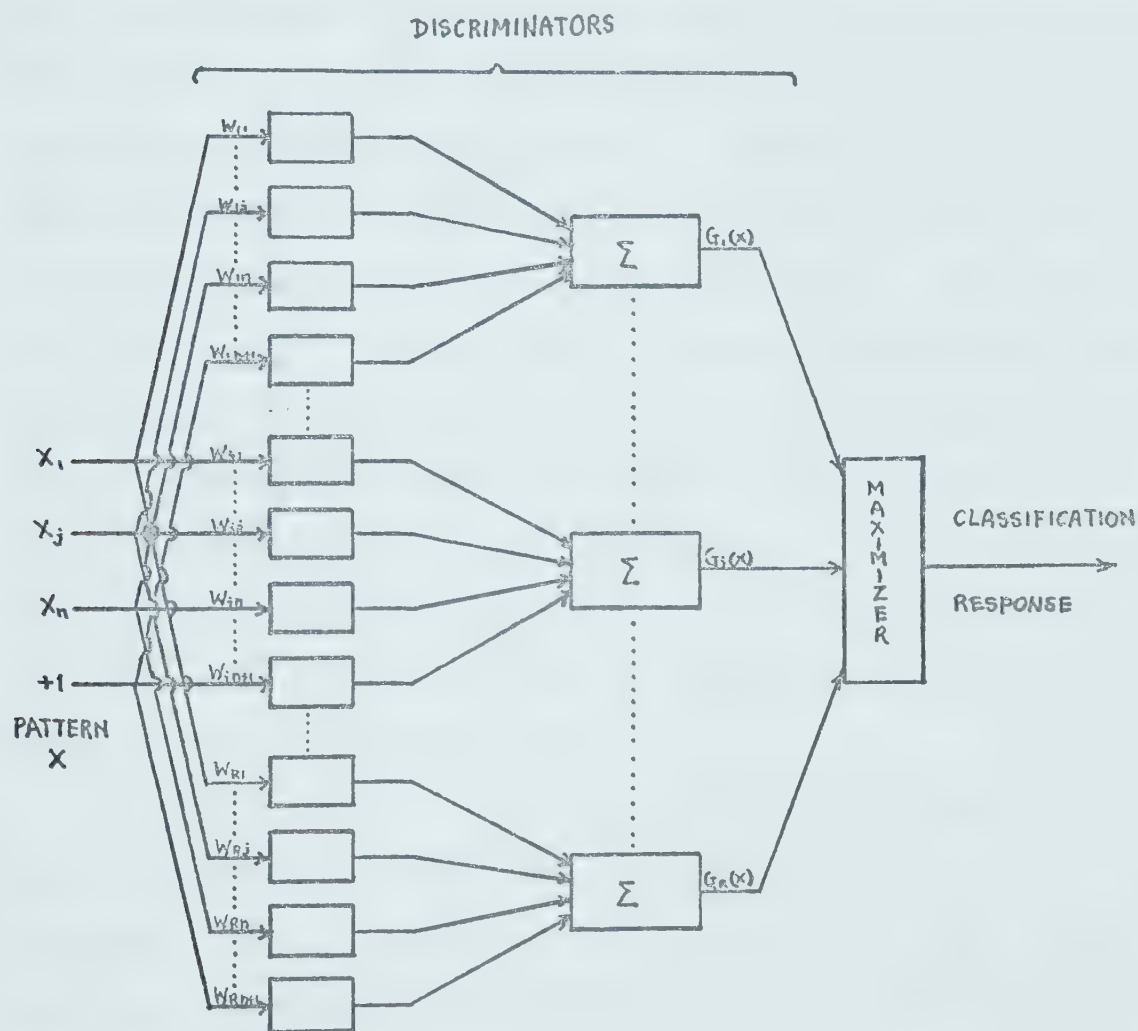


Fig. 6. A Linear Classifier.



## CHAPTER III

## FUZZY LOGIC

3.1. General.

In information theory we usually treat the uncertainty and imprecision of a problem through the concepts and methods of probability theory. However, in most real situations in information retrieval the source of imprecision is the absence of sharply defined criteria of class membership rather than the presence of random variables. Many classes are fuzzy rather than precise in nature. Objects need not necessarily either belong, or not belong, to a class; there may be intermediate grades of membership. To describe the degree with which an object belongs to a class, L. A. Zadeh proposed a multivalued logic with a possibly continuous infinity of truth values (31, 32, 33, 34). Zadeh's logic is based on the idea of fuzzy set.

A fuzzy set is a class with an unsharp boundary. It is an imprecisely defined class in which the transition from membership to non-membership is gradual rather than abrupt. There may be grades of membership intermediate between full membership and non-membership.

Let  $X$  be a collection of objects with each individual element denoted by  $x$ , that is  $X = \{x\}$ . Then a fuzzy set  $A$  in  $X$  can be characterized by a membership function  $\mu_A(x)$  which assigns to each element  $x$  in  $X$  a number in the closed interval between zero and one. The value of  $\mu_A(x)$  indicates the grade of membership of  $x$  in  $A$ . Thus, the nearer the



value of  $\mu_A(x)$  to unity, the higher will be the grade of membership of  $x$  in  $A$ .

The concept of fuzzy sets has proved to be relevant to a wide variety of problems related to information processing, information control, pattern recognition, system identification, artificial intelligence, and many other types of decision processes that involve incomplete or uncertain data. The idea of fuzzy set has also found application in feature extraction, which is the first stage of the proposed automatic classification system.

### 3.2. Basic Definitions of Fuzzy Sets.

Prior to using fuzzy logic in application to practical problems it is necessary to construct a mathematical framework for manipulation of fuzzy sets and the study of their properties. We shall begin the discussion of fuzzy sets with a number of basic definitions.

- 1) Two fuzzy sets  $A$  and  $B$  are said to be equal, written as  $A = B$ , if and only if  $\mu_A(x) = \mu_B(x)$  for all  $x$  in  $X$ .
- 2) A fuzzy set  $A$  is empty if and only if its membership function is identically zero for all  $x$  in  $X$ , that is,  $\mu_A(x) = 0$  for all  $x$  in  $X$ .
- 3) The complement of a fuzzy set  $A$  is a fuzzy set  $A'$  whose membership function is given by  $\mu_{A'}(x) = 1 - \mu_A(x)$ .
- 4) A fuzzy set  $A$  is contained in fuzzy set  $B$ , or equivalently,  $A$  is a subset of  $B$ , or  $A$  is smaller





than or equal to B, written as  $A \subseteq B$ , if and only if  $\mu_A(x) \leq \mu_B(x)$  for all  $x$  in  $X$ .

- 5) The union of two fuzzy sets A and B is denoted by  $A \cup B$  and is defined as the smallest fuzzy set that contains both A and B. The membership function of  $A \cup B$  is expressed by  $\mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)]$ , or equivalently,  $\mu_{A \cup B}(x) = \mu_A(x) \vee \mu_B(x)$ .
- 6) Similarly, the intersection of two fuzzy sets A and B is denoted by  $A \cap B$  and is defined as the largest fuzzy set contained in both A and B. The membership function of  $A \cap B$  is expressed by  $\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)]$ , or equivalently,  $\mu_{A \cap B}(x) = \mu_A(x) \wedge \mu_B(x)$ .

The intersection and union of two fuzzy sets A and B can be illustrated graphically as in Fig. 7.

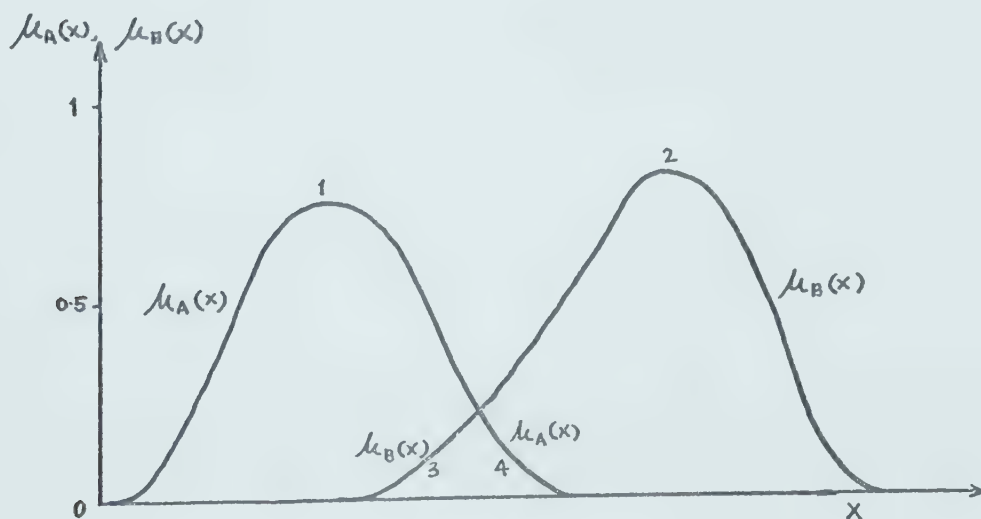


Fig. 7. Diagram Illustrating the Union and Intersection of Two Fuzzy Sets.

The membership function of union of fuzzy sets A and B



is comprised of curve segments 1 and 2, that of the intersection is comprised of 3 and 4.

The notions of union, intersection and complementation play important roles in fuzzy logic; they can be extended easily to many basic identities for fuzzy sets. Examples of a few are as follows:

$$\left. \begin{aligned} A \cup (B \cap C) &= (A \cup B) \cap C \\ A \cap (B \cup C) &= (A \cap B) \cup C \end{aligned} \right\} \text{Associative Law,}$$

$$\left. \begin{aligned} (A \cup B)' &= A' \cap B' \\ (A \cap B)' &= A' \cup B' \end{aligned} \right\} \text{DeMorgan's Law,}$$

$$\left. \begin{aligned} C \cap (A \cup B) &= (C \cap A) \cup (C \cap B) \\ C \cup (A \cap B) &= (C \cup A) \cap (C \cup B) \end{aligned} \right\} \text{Distributive Law.}$$

These, and many other similar equalities, can be readily established by showing that the corresponding relations for the membership functions of A, B and C are identities. For example,

$$(A \cup B)' = A' \cap B'$$

is equivalent to

$$1 - \max[\mu_A(x), \mu_B(x)] = \min[1 - \mu_A(x), 1 - \mu_B(x)].$$

Similarly,

$$C \cup (A \cap B) = (C \cup A) \cap (C \cup B)$$

is equivalent to



$$\max[\mu_c(x), \min\{\mu_A(x), \mu_B(x)\}] = \min[\max\{\mu_c(x), \mu_A(x)\}, \max\{\mu_c(x), \mu_B(x)\}].$$

One can interpret the intersection and union of fuzzy sets in terms of dropping a ball bearing through a network of pipes. Let  $B$  be a fuzzy set which is expressed in terms of a family of fuzzy sets  $A_1, A_2, \dots, A_n$  through the connections  $\vee$  and  $\wedge$ , and let  $\mu_i(x)$  be the membership function for fuzzy set  $A_i$ ,  $i = 1, 2, \dots, n$  with respect to  $x$ . Analogously, let  $P$  be a pipe whose passage clearance for a ball bearing  $x$  through it can be simulated by the resultant passage clearance for the same ball bearing  $x$  through a network of pipes,  $Q_1, Q_2, \dots, Q_n$  in series and parallel connections. If  $S_i(x)$  is the passage clearance for ball bearing  $x$  through pipe  $Q_i$ ,  $i = 1, 2, \dots, n$ , then  $\mu_i(x) \vee \mu_j(x)$  and  $\mu_i(x) \wedge \mu_j(x)$  correspond to parallel (or operation) and series (and operation) of  $S_i(x)$  and  $S_j(x)$  respectively as shown in Fig. 8.

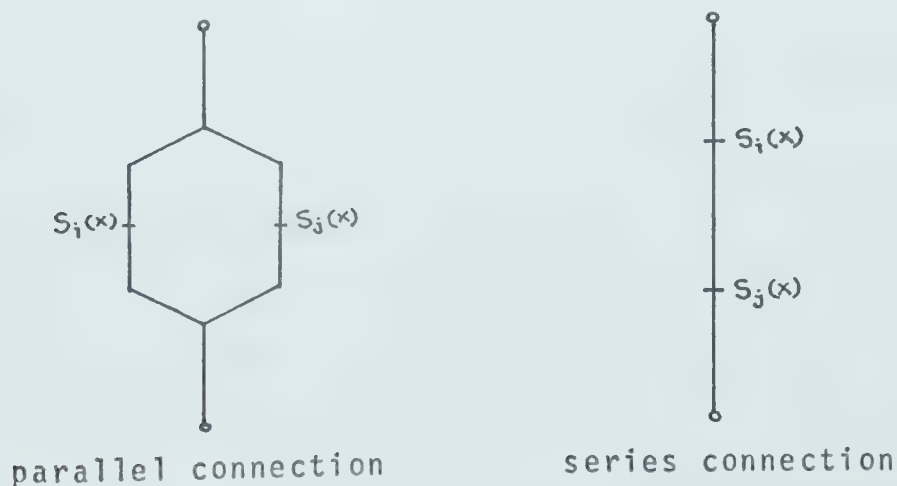


Fig. 8. Parallel and Series Connection of Pipes Simulating Union and Intersection of Two Fuzzy Sets.



Therefore, an expression involving  $A_1, A_2, \dots, A_n, \cup$  and  $\cap$  corresponds to a network of pipes  $Q_1, Q_2, \dots, Q_n$  which can be formed by the conventional synthesis techniques employed in switching theory. As an example,

$$C = [(A_1 \cap A_2) \cup A_3] \cup [(A_4 \cup A_5) \cap A_6]$$

can be denoted by the network of pipes as shown in Fig. 9.

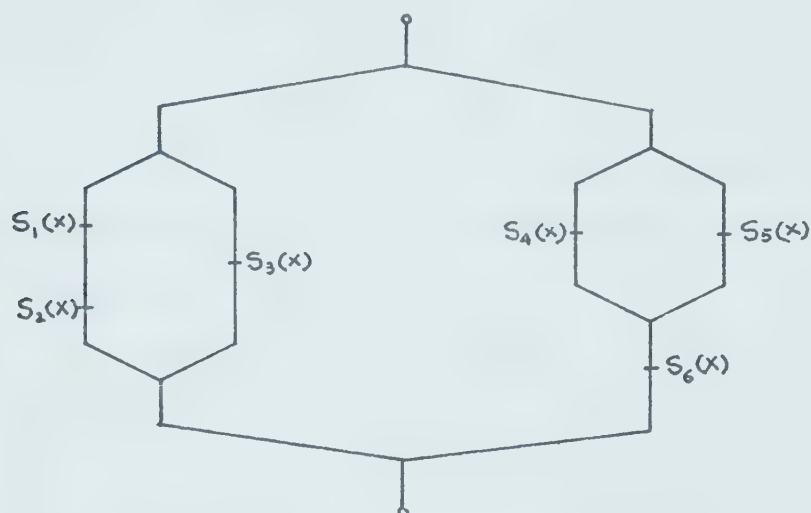


Fig. 9. A Network of Pipes Simulating

$$[(\mu_1(x) \cap \mu_2(x)) \cup \mu_3(x)] \cup [(\mu_4(x) \cup \mu_5(x)) \cap \mu_6(x)].$$

It may be noted that the passage of a ball bearing  $x$  through a pipe also depends on the diameter of the ball bearing. The whole network itself is equivalent to a single pipe whose passage clearance is of size  $S_p(x)$ .

### 3.3. Algebraic Operations on Fuzzy Sets.

Besides the operations of intersection and union, there are other ways of forming combinations of fuzzy sets and of relating them to one another.

- 1) The algebraic product of fuzzy sets  $A$  and  $B$  is





written as  $A \cdot B$  and can be defined in terms of the membership functions of  $A$  and  $B$  by the relation

$$\mu_{A \cdot B}(x) = \mu_A(x) \cdot \mu_B(x).$$

- 2) The algebraic sum of fuzzy sets  $A$  and  $B$  is denoted by  $A + B$  and is defined by

$$\mu_{A+B}(x) = \mu_A(x) + \mu_B(x)$$

provided that the sum of  $\mu_A(x) + \mu_B(x)$  is less than or equal to unity for all  $x$  in  $X$ .

- 3) The absolute difference of fuzzy sets  $A$  and  $B$  is written as  $|A - B|$  and is defined by

$$\mu_{|A-B|}(x) = |\mu_A(x) - \mu_B(x)|.$$

- 4) Let  $A$  and  $B$  be two arbitrary fuzzy sets. The convex combination of  $A$ ,  $B$  and a third fuzzy set  $\Lambda$  is denoted by  $(A, B; \Lambda)$  and can be defined as the linear combination of  $A$  and  $B$  in the form

$$(A, B; \Lambda) = \Lambda A + \Lambda' B$$

where  $\Lambda'$  is the complement of  $\Lambda$ . Expressing the relationship in terms of the membership functions, we have

$$\mu_{(A,B;\Lambda)}(x) = \mu_\Lambda(x) \mu_A(x) + [1 - \mu_\Lambda(x)] \mu_B(x).$$

A basic property of the convex combination of fuzzy sets  $A$ ,  $B$  and  $\Lambda$  is expressed by

$$A \cap B \subset (A, B; \Lambda) \subset A \cup B, \quad \text{for all } \Lambda.$$

This property is an immediate consequence of the inequalities:

$$\begin{aligned} \min[\mu_A(x), \mu_B(x)] &\leq \lambda \mu_A(x) + (1 - \lambda) \mu_B(x) \\ &\leq \max[\mu_A(x), \mu_B(x)], \quad \text{where } 0 \leq \lambda \leq 1. \end{aligned}$$



### 3.4. Fuzzy Relation.

The concept of relation has a natural extension to fuzzy sets. It plays an important role in the theory of fuzzy sets and their applications. The term "relation" can be defined as a set of ordered pairs (34). For example, the set of all ordered pairs of positive integers  $x$  and  $y$  such that  $x = y^2$  can be regarded as a relation between  $x$  and  $y$ . In terms of fuzzy sets, if  $X = \{x\}$  and  $Y = \{y\}$ , then a fuzzy relation  $R$  between  $X$  and  $Y$  is a fuzzy set  $F$  in the product space  $X \times Y$  such that  $F$  is characterized by a membership function  $\mu_R(x, y)$  which associates with each pair  $(x, y)$  its grade of membership in  $F$ . For example, the fuzzy relation,  $x \ll y$ , where  $x$  and  $y$  are both in space  $S$ , may be regarded as a fuzzy set  $B$  in  $S \times S$  or  $S^2$  such that the membership function of  $B$ ,  $\mu_B(x, y)$ , may take on the following representative values:

$$\mu_B(9, 10) = 0,$$

$$\mu_B(10, 100) = 0.6,$$

$$\mu_B(1, 1000) = 1, \text{ etc.}$$

The values of the membership function  $\mu_B(x, y)$  lie within the closed interval  $[0, 1]$  and it is referred to as the strength of the relation between  $x$  and  $y$ .

More generally, one can interpret an  $n$  - order fuzzy relation of fuzzy set  $X$  in space  $Y$  as a fuzzy set  $A$  in the product space  $Y^n$  with the membership function in the form  $\mu_A(x_1, x_2, \dots, x_i, \dots, x_n)$  where  $x_i$  is a member of  $X$ .

Before proceeding further, let us look at some of the



basic definitions that are related to fuzzy relations:

- 1) The domain of a fuzzy relation  $A$  is denoted by  $\text{dom } A$  and is a fuzzy set defined by

$$\mu_{\text{dom } A}(x) = \bigvee_y \mu_A(x, y), \quad x \in X,$$

where the supremum  $\bigvee$  is taken over all  $y$  in  $Y$ .

- 2) The range of a fuzzy relation  $A$  is denoted by  $\text{ran } A$  and is a fuzzy set defined by

$$\mu_{\text{ran } A}(y) = \bigvee_x \mu_A(x, y), \quad x \in X, y \in Y.$$

- 3) The height of a fuzzy set  $A$  is denoted by  $h(A)$  and is defined by

$$h(A) = \bigvee_x \bigvee_y \mu_A(x, y).$$

A fuzzy relation  $A$  is said to be subnormal if

$h(A) \leq 1$  and normal if  $h(A) = 1$ .

- 4) A fuzzy relation  $A$  is said to be contained in fuzzy relation  $B$  if

$$\mu_A(x, y) \leq \mu_B(x, y), \quad \text{for all } (x, y) \text{ in the product space } X \times Y.$$

The containment of  $A$  in  $B$  is expressed in the form  $A \subseteq B$ .

- 5) The union of fuzzy relations  $A$  and  $B$  is denoted by  $A + B$  and is defined by

$$\mu_{A+B}(x, y) = \max[\mu_A(x, y), \mu_B(x, y)]$$

where  $x \in X$  and  $y \in Y$ . Union may also be expressed by writing

$$\mu_{A+B}(x, y) = \mu_A(x, y) \vee \mu_B(x, y).$$

- 6) The intersection of two fuzzy relations  $A$  and  $B$  is denoted by  $A \cap B$  and is defined by

$$\mu_{A \cap B}(x, y) = \min[\mu_A(x, y), \mu_B(x, y)]$$

where  $x \in X$  and  $y \in Y$ . It may also be expressed



in the form

$$\mu_{A \cap B}(x, y) = \mu_A(x, y) \wedge \mu_B(x, y).$$

- 7) The product of two fuzzy relations A and B is denoted by  $A \cdot B$  and is defined by

$$\mu_{A \cdot B}(x, y) = \mu_A(x, y) \cdot \mu_B(x, y).$$

It may be noted that if A, B and C are any fuzzy relations from X to Y, then the identity

$$C(A + B) = CA + CB$$

will hold.

- 8) The complement of a fuzzy relation A is denoted by  $A'$  and is defined by

$$\mu_{A'}(x, y) = 1 - \mu_A(x, y).$$

- 9) The composition, or more specifically, the max - min composition, of two fuzzy relations A and B is denoted by  $B \circ A$  and is defined as a fuzzy relation whose membership function is related to those of A and B by

$$\mu_{B \circ A}(x, y) = \max_v \min[\mu_A(x, v), \mu_B(v, y)],$$

or equivalently,

$$\mu_{B \circ A}(x, y) = \bigvee_v [\mu_A(x, v) \wedge \mu_B(v, y)].$$

In the following section the max - min composition is used to define a similarity relation.

### 3.5. Similarity Relation.

It may be noted that the operation of max - min composition can be applied to a single fuzzy relation A to find the similarity relation of elements in a fuzzy set A. The similarity relation S between elements x and y in fuzzy





set  $A$  may be defined in terms of membership function as follows:

$$\mu_{A \circ A}(x, y) = \max_v \min[\mu_A(x, v), \mu_A(v, y)]$$

where  $x, y$  and  $v$  are all contained in  $A$ , and  $v$  ranges through all its  $n$  possible values where  $n$  is the number of elements in fuzzy set  $A$ .

A set of  $n$   $\min \mu_A(x, y)$  functions are selected between pairs of  $\mu_A(x, v)$  and  $\mu_A(v, y)$  and the  $\max \mu_A(x, y)$  is searched through the set of  $\min \mu_A(x, y)$  functions.

The  $n$  - order composition of  $A \circ A \circ \dots \circ A$  is denoted by  $A^n$ . If  $A$  is a finite set,  $\mu_A$  may be represented by a relation matrix whose  $(x, y)$ th element takes on the value of  $\mu_A(x, y)$ . The similarity relation matrix for a fuzzy set  $A$  is given by the max - min composition of the elements in a relation matrix  $A$ .

The concept of a similarity relation is a generalization of the concept of equivalence. Zadeh found that it is possible to adapt the well-developed theory of relations to situations which involve classes that do not have sharply defined boundaries. A similarity relation  $S$  is a fuzzy relation that is reflexive, symmetric, and transitive.

Let  $x, y$  be elements of a fuzzy set  $A$ , and let  $\mu_S(x, y)$  denotes the grade of membership of the ordered pair  $(x, y)$  in  $S$ . Then  $S$  is a similarity relation in  $A$  if and only if for all  $x, y$  and  $v$  in  $A$ :

$$1) \mu_S(x, x) = 1 \quad \text{for all } x \text{ in dom } S \text{ (reflexive).}$$

[3.1]



$$2) \mu_s(x, y) = \mu_s(y, x) \quad \text{for all } x \text{ and } y \text{ in dom } S$$

(symmetry). [3.2]

$$3) S \supseteq S \circ S \text{ (max - min transitivity) or more specifically}$$

$$\mu_s(x, y) \geq \bigvee [\mu_s(x, v) \wedge \mu_s(v, y)] \quad [3.3]$$

where  $\bigvee$  and  $\wedge$  denote max and min respectively.

An example of a fuzzy relation matrix having similarity relations is shown in Fig. 10.

$$\begin{bmatrix} 1 & 0.3 & 0.8 & 0.5 & 0.3 \\ 0.3 & 1 & 0.6 & 0.2 & 0.5 \\ 0.8 & 0.6 & 1 & 0.1 & 0.4 \\ 0.5 & 0.2 & 0.1 & 1 & 0.2 \\ 0.3 & 0.5 & 0.4 & 0.2 & 1 \end{bmatrix}$$

Fig. 10. Relation Matrix Having Similarity Relations.

Let  $x_{i_1}, x_{i_2}, \dots, x_{i_k}$  be  $k$  points in  $X$  such that  $\mu(x_{i_1}, x_{i_2}), \dots, \mu(x_{i_{k-1}}, x_{i_k})$  are all greater than zero. Then the sequence  $C = (x_{i_1}, x_{i_2}, \dots, x_{i_k})$  will be said to be a chain from  $x_{i_1}$  to  $x_{i_k}$  with the strength of the chain defined as the strength  $\mu(\ )$  of its weakest link, that is,

$$\text{Strength of } (x_{i_1}, x_{i_2}, \dots, x_{i_k}) = \min[\mu(x_{i_1}, x_{i_2}), \mu(x_{i_2}, x_{i_3}), \dots, \mu(x_{i_{k-1}}, x_{i_k})].$$

From the definition of the max - min composition, it follows that the  $(i, j)$ th elements of  $S^n$ ,  $n = 1, 2, \dots$ , is the strength of the strongest chain of length  $n$  from  $x_i$  to  $x_j$ . Thus, the transitivity condition may be stated in words as follows:



Strength of  $S$  between  $x_i, x_j =$

Strength of the strongest chain from  $x_i$  to  $x_j$ .

It may be noted that if  $X$  has  $m$  elements, then any chain  $C$  of length  $k > m$  from  $x_{i_1}$  to  $x_{i_k}$  must necessarily have loops (the presence of one or more elements in  $X$  appear more than once in the chain  $C = (x_{i_1}, x_{i_2}, \dots, x_{i_k})$ ). If the loops are removed, the resulting chain  $\tilde{C}$  of length  $\leq m$  will have at least the same strength as  $C$ . Consequently, for any elements  $x_i, x_j$  in  $X$ , we can assert that:

Strength of the strongest chain from  $x_i$  to  $x_j =$

Strength of the strongest chain of length  $\leq m$  from  $x_i$  to  $x_j$ .

### 3.6. Feature Extraction Based on Fuzzy Relation.

In 1971 S. Tamura, S. Higuchi and K. Tanaka proposed a method of classifying patterns using fuzzy relation to measure the similarity between each pair of patterns taken from the population of patterns to be classified (19). The similitude between any two patterns is calculated using the max - min composition of a fuzzy relation. The similitude is extended to  $n$  - step such that the complete similitude between two patterns is achieved.

Let  $X$  be a set of patterns, the fuzzy relation  $A$  on  $X$  is characterized by its membership function  $\mu_A(x, y) \in [0, 1]$ , for all  $x, y \in X$ . The one - step fuzzy relation  $\mu_1(x, y)$  satisfies two conditions:

- 1)  $\mu_1(x, x) = 1$  for all  $x$  in  $X$ ,
- 2)  $\mu_1(x, y) = \mu_1(y, x)$  for all  $x, y$  in  $X$ .

Condition 1 implies that  $x$  is exactly the same pattern



as  $x$ , and condition 2 implies that the fuzzy relation is symmetric.

If  $\mu_1(x, y)$  is known for each of the pairs of patterns in  $X$ , then the  $n$  - step fuzzy relation  $\mu_n(x, y)$  may be defined by the equation:

$$\mu_n(x, y) = \max_{x_1, x_2, \dots, x_{n-1} \in X} \min[\mu_1(x, x_1), \mu_1(x_1, x_2), \dots, \mu_1(x_{n-1}, y)]$$

where  $n = 2, 3, \dots$ .

Similarly, the  $(n+1)$  - step fuzzy relation is given by

$$\mu_{n+1}(x, y) = \max_{x_1, x_2, \dots, x_{n-1}, x_n \in X} \min[\mu_1(x, x_1), \mu_1(x_1, x_2), \dots, \mu_1(x_{n-1}, x_n), \mu_1(x_n, y)].$$

It may be noted that

$$\begin{aligned} \mu_{n+1}(x, y) &= \max_{x_1, x_2, \dots, x_{n-1}, x_n \in X} \min[\mu_1(x, x_1), \mu_1(x_1, x_2), \dots, \mu_1(x_{n-1}, x_n), \mu_1(x_n, y)] \\ &\geq \max_{x_1, x_2, \dots, x_{n-1} \in X} \min[\mu_1(x, x_1), \mu_1(x_1, x_2), \dots, \mu_1(x_{n-1}, y), \mu_1(y, y)] \\ &= \mu_n(x, y). \end{aligned}$$

Therefore,  $0 \leq \mu_1(x, y) \leq \mu_2(x, y) \leq \dots \leq \mu_n(x, y) \leq$

$\mu_{n+1}(x, y) \leq \dots \leq 1$ , and the value of the complete

similitude  $\mu(x, y)$  is in the closed interval between zero

and one such that

$$\mu(x, y) = \lim_{n \rightarrow \infty} \mu_n(x, y).$$

As an example, let  $X = \{x_1, x_2, x_3, x_4\}$  with  $\mu_1(x_i, x_j)$ ,  $i, j = 1, 2, 3, 4$  given as in Figs. 11 and 12.





	$x_1$	$x_2$	$x_3$	$x_4$
$x_1$	1			
$x_2$	0.2	1		
$x_3$	0	0.5	1	
$x_4$	0.6	0.3	0	1

Fig. 11. 1 - step Fuzzy Relation Matrix of Elements  $\mu_1(x_i, x_j)$ 's.

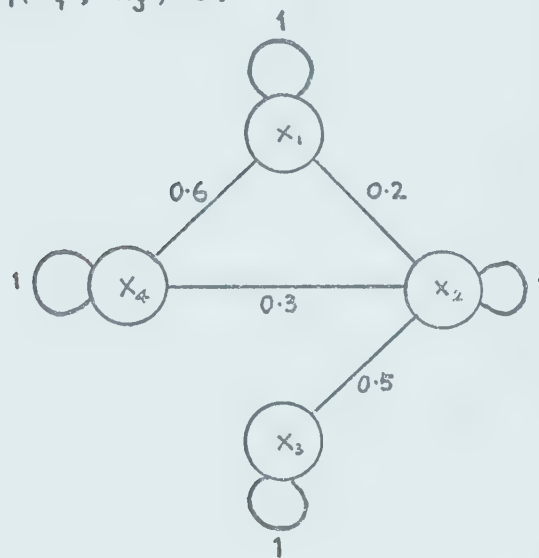


Fig. 12. 1 - step Fuzzy Relation Diagram for  $\mu_1(x_i, x_j)$ 's.

The complete fuzzy relation  $\mu(x_i, x_j) = \mu_1(x_i, x_j)$  is shown in Figs. 13 and 14.

	$x_1$	$x_2$	$x_3$	$x_4$
$x_1$	1			
$x_2$	0.3	1		
$x_3$	0.2	0.5	1	
$x_4$	0.6	0.3	0.5	1

Fig. 13. 2 - step Fuzzy Relation Matrix of Elements  $\mu_2(x_i, x_j)$ 's.



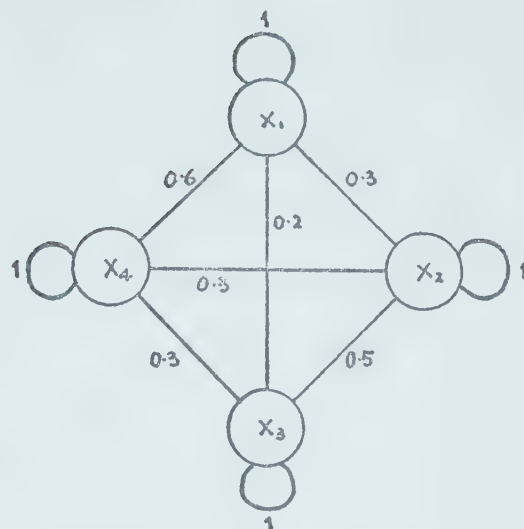


Fig. 14. Complete Fuzzy Relation Diagram for  $\mu(x_i, x_j)$ 's.

When the set of patterns  $X$  has a finite number of elements, it is sometimes convenient to represent the fuzzy relation between elements in matrix form. We represent the fuzzy relation matrix  $F$  as,

$$F = \|\mu_i(x_i, x_j)\|$$

where  $i, j = 1, 2, \dots, n$ ,  $x_i$  and  $x_j$  are in  $X$ , and  $n$  is the number of elements in  $X$ .

For a fuzzy relation matrix  $A$ , we denote the  $(i, j)$ th entry by  $a_{ij}$  where  $0 \leq a_{ij} \leq 1$ . We may then define:

- 1)  $A \leq B$  if and only if  $a_{ij} \leq b_{ij}$  for all  $i, j$  in  $n$ .
- 2)  $I = \|m_{ij}\|$ ,  
where  $m_{ij} = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases}$
- 3)  $C = A \circ B$  if and only if  $c_{ij} = \max_k \min(a_{ik}, b_{kj})$ .
- 4)  $A^{m+1} = A^m \circ A$ .
- 5)  $A^0 = I$ .
- 6)  $C = \max(A, B)$  if and only if  $c_{ij} = \begin{cases} a_{ij} & \text{if } a_{ij} \geq b_{ij} \\ b_{ij} & \text{otherwise.} \end{cases}$

Thus, all diagonal elements of any fuzzy relation matrix



$F$  are equal to unity and  $I < F < F^2 < \dots < F^{n-1} = F^n = \dots = F^\infty$ . The  $(i, j)$ th entry of  $F^k$  is the value of  $\mu_k(x_i, x_j)$ . Hence, the complete fuzzy relation  $\mu(x_i, x_j)$  may be calculated rather easily and quickly by successive application of  $F^k \circ F^k = F^{2k}$  where  $k = 1, 2, \dots$ .

As an example, let

$$F = \begin{bmatrix} 1 & 0.8 & 0 & 0.1 & 0.2 \\ 0.8 & 1 & 0.4 & 0 & 0.9 \\ 0 & 0.4 & 1 & 0 & 0 \\ 0.1 & 0 & 0 & 1 & 0.5 \\ 0.2 & 0.9 & 0 & 0.5 & 1 \end{bmatrix}$$

then,  $F^2 = F \circ F$

$$F^2 = \begin{bmatrix} 1 & 0.8 & 0.4 & 0.2 & 0.8 \\ 0.8 & 1 & 0.4 & 0.5 & 0.9 \\ 0.4 & 0.4 & 1 & 0 & 0.4 \\ 0.2 & 0.5 & 0 & 1 & 0.5 \\ 0.8 & 0.9 & 0.4 & 0.5 & 1 \end{bmatrix}$$

and  $F^4 = F^2 \circ F^2$

$$F^4 = \begin{bmatrix} 1 & 0.8 & 0.4 & 0.5 & 0.8 \\ 0.8 & 1 & 0.4 & 0.5 & 0.9 \\ 0.4 & 0.4 & 1 & 0.4 & 0.4 \\ 0.5 & 0.5 & 0.4 & 1 & 0.5 \\ 0.8 & 0.9 & 0.4 & 0.5 & 1 \end{bmatrix}.$$

In fact  $F^3 = F^4 = F^5 = \dots = F^\infty$ , thus,  $I < F < F^2 < F^3 < \dots < F^\infty$ .



## CHAPTER IV

### KARHUNEN - LOÈVE EXPANSION

#### 4.1. General.

Before an input pattern can be fed into a classifier, there are two problems that must be solved. The first is a sensing problem; the designer has to decide what is to be measured from the input patterns and how to associate measured quantities with individual characteristics. Usually, the raw data being measured is of high dimension, it contains much redundant and insignificant information that contributes little to the recognition process. The second problem is concerned with feature selection and preprocessing; the preprocessor attempts to select discriminatory characteristic features or attributes from the input pattern space. Feature selection is of major importance in pattern recognition systems, it reduces the dimensionality of the input measurement vector and in turn greatly reduces the computation time required in the recognition process. The present chapter is devoted to discussion of a possible solution to the second problem; discussion of the first problem is deferred until Chapter V.

The selection of features can generally be divided into two phases, namely, preselection and postselection (35). Preselection requires no knowledge of class samples. Features of the input pattern that are known to be ineffective in discriminating between individual objects, or are judged to be useless from a general knowledge about the





nature of the desired class, will be eliminated in this phase. It is usually performed in conjunction with feature extraction which is the first stage of any automatic pattern recognition system. On the basis of the collections of sample patterns at hand, postselection selects from the sample patterns those features which are most effective in distinguishing samples of one class from those of another. A number of methods have been proposed for postselection, and the author is particularly interested in a method derived from an optimal expansion known as the generalized Karhunen - Loève expansion.

#### 4.2. The Generalized Karhunen - Loève Expansion.

An optimal feature selection and feature ordering procedure may be developed based on the generalized Karhunen - Loève expansion. The procedure depends on use of an orthogonal transformation of coordinates in the representation space in order to obtain the optimal coordinate system with weights sharply concentrated in a few coordinates. Application of the Karhunen - Loève expansion to feature selection was first suggested by S. Watanabe in 1965 (27) and further developed by K. S. Fu and T. Y. Chien in 1967 (28). This feature selection and ordering scheme has the important characteristic of not requiring the computation of the probability of misrecognition, or complete knowledge of the probability distribution of the input pattern. It is applicable to any data processing system where a reduction of dimensionality or compression of information is desired prior to subsequent processing. The basis of the scheme is



a preweighting of features according to their relative importance in description of the input patterns. Relative importance is meant in the sense of carrying more information regarding the discrimination of pattern classes so that use of only a finite number of these features introduces a relatively small error.

#### 4.2.1. Derivation of the Generalized Karhunen - Loève Expansion.

Consider observation of a stochastic process  $\{X(t), 0 \leq t \leq T\}$  over a period of time  $(0, T)$ , the observed random function  $\{X(t), 0 \leq t \leq T\}$  being generated from  $m$  possible stochastic processes  $\{X_i(t), 0 \leq t \leq T\}$ ,  $i = 1, 2, \dots, m$  and  $m \geq 2$ , corresponding to  $m$  pattern classes. Let  $P_i$  be the probability that the  $i$ th process occurs, and suppose  $\sum_{i=1}^m P_i = 1$ . We wish to express the random function  $X_i(t)$  in the form:

$$X_i(t) = \sum_{k=1}^{\infty} V_{ik} \phi_k(t) \quad \text{for all } t \in (0, T) \text{ and} \\ i = 1, 2, \dots, m, \quad [4.1]$$

where the  $V_{ik}$  's are random coefficients that satisfy the relation  $E(V_{ik}) = 0$  (achieved by centralizing all the random functions). The set  $\{\phi_k(t)\}$  is a set of deterministic orthonormal coordinate functions over the range  $(0, T)$ , that is,

$$\int_0^T \phi_k(t) \phi_{\ell}^*(t) dt = \delta_{k\ell} \quad [4.2]$$

where the  $*$  indicates the complex conjugate and  $\delta_{k\ell}$  is the



Kronecker delta function equal to one if  $k = \ell$  and equal to zero otherwise.

We define a covariance function  $K(t, s)$  for the  $m$  stochastic processes as follows:

$$K(t, s) = \sum_{i=1}^m P_i E[X_i(t) X_i^*(s)]. \quad [4.3]$$

Substituting [4.1] into [4.3] gives

$$\begin{aligned} K(t, s) &= \sum_{i=1}^m P_i E\left[\left\{\sum_{k=1}^{\infty} V_{ik} \phi_k(t)\right\}\left\{\sum_{\ell=1}^{\infty} V_{i\ell}^* \phi_{\ell}^*(s)\right\}\right] \\ &= \sum_{k, \ell=1}^{\infty} \phi_k(t) \phi_{\ell}^*(s) \sum_{i=1}^m P_i E(V_{ik} V_{i\ell}^*). \end{aligned} \quad [4.4]$$

If furthermore, the random coefficients  $V_{ik}$ 's are chosen to satisfy the conditions:

$$\begin{aligned} \sum_{i=1}^m P_i E(V_{ik} V_{i\ell}^*) &= \sum_{i=1}^m P_i \text{Var}(V_{ik}) = \sigma_k^2 & \text{if } k = \ell, \\ \text{and } \sum_{i=1}^m P_i E(V_{ik} V_{i\ell}^*) &= 0 & \text{if } k \neq \ell, \end{aligned} \quad [4.5]$$

then the covariance function  $K(t, s)$  can be expressed in the form:

$$K(t, s) = \sum_{k=1}^{\infty} \sigma_k^2 \phi_k(t) \phi_k^*(s). \quad [4.6]$$

In other words, if the expansion in [4.1] exists for the random function  $X_i(t)$ , and the random coefficients  $V_{ik}$ 's satisfy the conditions in [4.5], then the covariance function  $K(t, s)$  can be expressed in the form as shown in [4.6].

Furthermore, from [4.6] it follows that

$$\int_0^T K(t, s) \phi_k(s) ds = \int_0^T \left[ \sum_{\ell=1}^{\infty} \sigma_{\ell}^2 \phi_{\ell}(t) \phi_{\ell}^*(s) \right] \phi_k(s) ds.$$

Then, if the summation and integration may be interchanged,



then

$$\begin{aligned}
 \int_0^T K(t, s) \varphi_k(s) ds &= \sum_{\ell=1}^{\infty} \sigma_{\ell}^2 \varphi_{\ell}(t) \int_0^T \varphi_k(s) \varphi_{\ell}^{*}(s) ds \\
 &= \sum_{\ell=1}^{\infty} \sigma_{\ell}^2 \varphi_{\ell}(t) \delta_{k\ell} \\
 &= \sigma_k^2 \varphi_k(t).
 \end{aligned}
 \tag{4.7}$$

Therefore,  $\{\sigma_k^2\}$  and  $\{\varphi_k(t)\}$  satisfy the integral equation defined in [4.7]. The expansion in [4.1], whose orthogonal coordinate functions  $\{\varphi_k(t)\}$  are determined by [4.7] through the covariance function  $K(t, s)$  is called the generalized Karhunen - Loève expansion.

In the terminology used to treat integral equations, the  $\{\varphi_k(t)\}$  are the characteristic functions, or eigenfunctions, and the  $\{\sigma_k^2\}$  are the characteristic values, or eigenvalues, of the covariance function  $K(t, s)$ .

#### 4.3. Optimal Properties of the Generalized Karhunen - Loève Expansion.

Fu and Chien have remarked that there are two optimal properties for the generalized Karhunen - Loève expansion (28), namely,

- 1) The expansion minimizes the mean square error that results by selecting a finite number of terms in the infinite series of expansion.
- 2) The expansion minimizes the entropy function defined over the variances of the coordinate coefficients in the expansion.

##### 4.3.1. Derivation of the First Property.

Let  $\{\varphi_k(t)\}$  be a set of arbitrary orthonormal





coordinate functions and let [4.1] be rewritten as:

$$X_i(t) = \sum_{k=1}^n v_{ik} \varphi_k(t) + R_{in}(t), \quad i = 1, 2, \dots, m, \quad [4.8]$$

where  $R_{in}(t)$  is the remainder when the expansion terminates at  $k = n$ . Define the expected value of the square of the modulus of the remainders by  $\sum_{i=1}^m P_i E[|R_{in}(t)|^2]$ . We wish to find the coordinate functions which give the best approximation to the random functions  $X_i(t)$ , in the sense that among all possible expansions having the same number of the terms, the particular choice of coordinate functions minimizes

$\sum_{i=1}^m P_i E[|R_{in}(t)|^2]$ . Writing out the expansion one obtains,

$$\begin{aligned} \sum_{i=1}^m P_i E[|R_{in}(t)|^2] &= \sum_{i=1}^m P_i E[R_{in}(t) R_{in}^*(t)] \\ &= \sum_{i=1}^m P_i E\left\{X_i(t) - \sum_{k=1}^n v_{ik} \varphi_k(t) \right\} \\ &\quad \left\{X_i^*(t) - \sum_{\ell=1}^n v_{i\ell}^* \varphi_{\ell}^*(t)\right\} \\ &= \sum_{i=1}^m P_i E\left[X_i(t) X_i^*(t) - \sum_{k=1}^n v_{ik} \varphi_k(t) X_i^*(t) \right. \\ &\quad \left. - \sum_{\ell=1}^n v_{i\ell}^* \varphi_{\ell}^*(t) X_i(t) + \sum_{k=1}^n v_{ik} \varphi_k(t) \sum_{\ell=1}^n v_{i\ell}^* \varphi_{\ell}^*(t)\right] \\ &= \sum_{i=1}^m P_i \left\{E[|X_i(t)|^2] - \sum_{k=1}^n \varphi_k(t) E[v_{ik} X_i^*(t)] \right. \\ &\quad \left. - \sum_{\ell=1}^n \varphi_{\ell}^*(t) E[v_{i\ell}^* X_i(t)] \right. \\ &\quad \left. + \sum_{k=1}^n \sum_{\ell=1}^n E(v_{ik} v_{i\ell}^*) \varphi_k(t) \varphi_{\ell}^*(t)\right\} \\ &= \sum_{i=1}^m P_i E[|X_i(t)|^2] \\ &\quad + \sum_{\ell=1}^n \sum_{k=1}^n \sum_{i=1}^m P_i E(v_{ik} v_{i\ell}^*) \varphi_k(t) \varphi_{\ell}^*(t) \\ &\quad - \sum_{k=1}^n \varphi_k(t) \sum_{i=1}^m P_i E[v_{ik} X_i^*(t)] \\ &\quad - \sum_{\ell=1}^n \varphi_{\ell}^*(t) \sum_{i=1}^m P_i E[v_{i\ell}^* X_i(t)]. \quad [4.9] \end{aligned}$$

Substituting [4.1] and [4.5] in terms of the generalized Karhunen - Loève expansion, we have



$$\begin{aligned}
\sum_{i=1}^m P_i E[V_{ik} X_i^*(t)] &= \sum_{i=1}^m P_i E[V_{ik} \{ \sum_{\ell=1}^{\infty} V_{i\ell} \varphi_{\ell}^*(t) \}] \\
&= [\sum_{\ell=1}^{\infty} \sum_{i=1}^m P_i E(V_{ik} V_{i\ell})] \varphi_{\ell}^*(t) \\
&= \sigma_k^2 \varphi_k^*(t).
\end{aligned} \tag{4.10}$$

Similarly,

$$\sum_{i=1}^m P_i E[V_{ik}^* X_i(t)] = \sigma_k^2 \varphi_k(t). \tag{4.11}$$

Substituting [4.10] and [4.11] into [4.9] gives

$$\begin{aligned}
\sum_{i=1}^m P_i E[|R_{in}(t)|^2] &= \sum_{i=1}^m P_i E[|X_i(t)|^2] + \sum_{k=1}^n \sigma_k^2 \varphi_k(t) \varphi_k^*(t) \\
&\quad - \sum_{k=1}^n \sigma_k^2 \varphi_k^*(t) \varphi_k(t) - \sum_{k=1}^n \sigma_k^2 \varphi_k(t) \varphi_k^*(t) \\
&= \sum_{i=1}^m P_i E[|X_i(t)|^2] + \sum_{k=1}^n \sigma_k^2 [|\varphi_k(t)|^2 \\
&\quad - \varphi_k(t) \varphi_k^*(t) - \varphi_k(t) \varphi_k(t)].
\end{aligned} \tag{4.12}$$

It can be shown that

$$\begin{aligned}
|\varphi_k(t)|^2 - \varphi_k(t) \varphi_k^*(t) - \varphi_k(t) \varphi_k(t) &= |\varphi_k(t) - \varphi_k(t)|^2 \\
&\quad - |\varphi_k(t)|^2.
\end{aligned}$$

Therefore, [4.12] can be rewritten as

$$\begin{aligned}
\sum_{i=1}^m P_i E[|R_{in}(t)|^2] &= \sum_{i=1}^m P_i E[|X_i(t)|^2] - \sum_{k=1}^n \sigma_k^2 |\varphi_k(t)|^2 \\
&\quad + \sum_{k=1}^n \sigma_k^2 |\varphi_k(t) - \varphi_k(t)|^2.
\end{aligned}$$

Obviously, the minimum value of  $\sum_{i=1}^m P_i E[|R_{in}(t)|^2]$  is attained at  $\varphi_k(t) = \varphi_k(t)$  where  $\varphi_k(t)$  is the generalized Karhunen - Loève coordinate function defined in [4.7]. Thus

$$\min \left\{ \sum_{i=1}^m P_i E[|R_{in}(t)|^2] \right\} = \sum_{i=1}^m P_i E[|X_i(t)|^2] - \sum_{k=1}^n \sigma_k^2 |\varphi_k(t)|^2.$$

It may be noted that the necessary conditions which allow the expansion of [4.1] to be the generalized Karhunen - Loève



expansion defined in [4.7] are

$$\sum_{i=1}^m P_i E(V_{ik} V_{il}^*) = \begin{cases} \sigma_k^2 & \text{if } k = l, \\ 0 & \text{if } k \neq l. \end{cases}$$

The relation implies that the random coefficients between each pair of coordinate functions among all classes of stochastic processes should be uncorrelated. It is noted, however, that the random coefficients between each pair of coordinate functions for a single class should not be uncorrelated.

#### 4.3.2. Derivation of the Second Property.

Let  $X_i(t)$  be square integrable and normalized such that

$$\int_0^T |X_i(t)|^2 dt = 1, \quad t \in (0, T) \text{ and } i = 1, 2, \dots, m. \quad [4.13]$$

Then from [4.1] we can show that  $\sum_{k=1}^{\infty} |V_{ik}|^2 = 1$ . The proof is as follows:

$$\begin{aligned} |X_i(t)|^2 &= X_i(t) X_i^*(t) \\ &= \left[ \sum_{k=1}^{\infty} V_{ik} \phi_k(t) \right] \left[ \sum_{l=1}^{\infty} V_{il} \phi_l^*(t) \right] \\ &= \sum_{k,l=1}^{\infty} V_{ik} V_{il}^* \phi_k(t) \phi_l^*(t). \end{aligned} \quad [4.14]$$

Substituting [4.14] into [4.13], we have

$$\begin{aligned} \int_0^T |X_i(t)|^2 dt &= \int_0^T \sum_{k,l=1}^{\infty} V_{ik} V_{il}^* \phi_k(t) \phi_l^*(t) dt \\ &= \sum_{k,l=1}^{\infty} V_{ik} V_{il}^* \left[ \int_0^T \phi_k(t) \phi_l^*(t) dt \right] \\ &= \sum_{k,l=1}^{\infty} V_{ik} V_{il}^* \delta_{kl} \\ &= \sum_{k=1}^{\infty} |V_{ik}|^2. \end{aligned} \quad [4.15]$$



Therefore, from [4.13] and [4.15], we may conclude that

$\sum_{k=1}^{\infty} |V_{ik}|^2 = 1$ . If we define  $\sigma_k^2$  for each coordinate function  $\phi_k(t)$ ,  $k = 1, 2, \dots$  as

$$\begin{aligned}\sigma_k^2 &= \sum_{i=1}^m P_i E |V_{ik}|^2 \\ &= \rho_k\end{aligned}$$

where the  $\sigma_k^2$ 's are the eigenvalues of the integral equation defined in [4.7], then

$$\begin{aligned}\sum_{k=1}^{\infty} \rho_k &= \sum_{k=1}^{\infty} \sum_{i=1}^m P_i E |V_{ik}|^2 \\ &= \sum_{i=1}^m P_i \sum_{k=1}^{\infty} E |V_{ik}|^2 \\ &= \sum_{i=1}^m P_i \\ &= 1.\end{aligned}$$

It may be noted that  $\rho_k \geq 0$ , therefore, the  $\rho_k$ 's form a probability distribution on the set of generalized Karhunen - Loève coordinate functions  $\{\phi_k(t)\}$ .

Now define an entropy function for the  $\rho_k$ 's of the  $\{\phi_k(t)\}$  as

$$H[\{\phi_k(t)\}] = - \sum_{k=1}^{\infty} \rho_k \log_e \rho_k. \quad [4.16]$$

If the  $\rho_k$ 's are ordered such that

$$\rho_1 \geq \rho_2 \geq \dots \geq \rho_k \geq \rho_{k+1} \geq \dots$$

then for any other similarly ordered  $\lambda_k$ 's associated with any arbitrary set of coordinate functions  $\{\psi_k(t)\}$  we have

$$\sum_{k=1}^n \rho_k \geq \sum_{k=1}^n \lambda_k. \quad [4.17]$$





Thus in terms of entropy

$$-\sum_{k=1}^{\infty} \rho_k \log_e \rho_k \leq -\sum_{k=1}^{\infty} \lambda_k \log_e \lambda_k \quad [4.18]$$

and

$$H[\{\phi_k(t)\}] = \min_{\{\varphi_k(t)\}} H[\{\varphi_k(t)\}]. \quad [4.19]$$

#### 4.4. Discrete Equivalent of the Generalized Karhunen - Loève Expansion. (28)

Suppose, instead of continuously observing a random function  $X_i(t)$  over a period of time, sampled measurements are taken from the random function in the following form:

$$X_i = (X_{i1}, X_{i2}, \dots, X_{ik}), \quad i = 1, 2, \dots, m,$$

where each  $X_i$  is a random vector of  $k$  components ( $k$  is finite). The desired expansion then becomes

$$X_{ij} = \sum_{n=1}^{\infty} V_{in} u_{nj}, \quad \begin{aligned} i &= 1, 2, \dots, m, \\ j &= 1, 2, \dots, k, \end{aligned} \quad [4.20]$$

where the  $V_{in}$ 's are the random coefficients and  $u_{nj}$  is the  $j$ th component of the  $n$ th orthonormal coordinate vector in a set of orthonormal coordinate vectors  $\{\phi_n\}$  which is analogous to the set of orthogonal coordinate functions  $\{\phi_k(t)\}$  in the continuous case. If we define the discrete analog of the covariance function  $K(t, s)$  for  $m$  stochastic processes as

$$\sum_{i=1}^m P_i E(X_{it}, X_{is}^*) \quad \text{where } t, s = 1, 2, \dots, k,$$

then



$$\begin{aligned}
K(t, s) &= \sum_{i=1}^m P_i E(X_{it}, X_{is}^*) \\
&= \sum_{i=1}^m P_i E\left\{\left(\sum_{n=1}^{\infty} V_{in} u_{nt}\right)\left(\sum_{\ell=1}^{\infty} V_{i\ell}^* u_{\ell s}^*\right)\right\} \\
&= \sum_{n=1}^{\infty} \sum_{\ell=1}^{\infty} u_{nt} u_{\ell s}^* \sum_{i=1}^m P_i E(V_{in} V_{i\ell}^*) \\
&= \sum_{\ell=1}^{\infty} \sigma_{\ell}^2 u_{\ell t} u_{\ell s}^*. \tag{4.21}
\end{aligned}$$

Furthermore, by the orthonormality of the coordinate vector, we have

$$\begin{aligned}
\sum_{s=1}^p K(t, s) u_{ns} &= \sum_{s=1}^p \left(\sum_{\ell=1}^{\infty} \sigma_{\ell}^2 u_{\ell t} u_{\ell s}^*\right) u_{ns} \\
&= \sum_{\ell=1}^{\infty} \sigma_{\ell}^2 u_{\ell t} \sum_{s=1}^p u_{\ell s}^* u_{ns} \\
&= \sum_{\ell=1}^{\infty} \sigma_{\ell}^2 u_{\ell t} \delta_{\ell n} \\
&= \sigma_n^2 u_{nt}. \tag{4.22}
\end{aligned}$$

Therefore, the generalized Karhunen - Loève expansion for the discrete case becomes

$$\begin{aligned}
X_{ij} &= \sum_{n=1}^{\infty} V_{in} u_{nj} \quad i = 1, 2, \dots, m, \\
&\quad j = 1, 2, \dots, k,
\end{aligned}$$

where  $u_{nj}$  is the  $j$ th component of the  $n$ th orthonormal coordinate vector satisfying [4.22]. The random coefficient  $V_{in}$  is determined for each  $n$  by the equation:

$$V_{in} = \sum_{j=1}^k X_{ij} u_{nj}^* \quad i = 1, 2, \dots, m. \tag{4.23}$$

It may be noted that the coordinate vectors  $u_{nj}$ 's of the generalized Karhunen - Loève expansion are essentially the eigenvectors determined from the covariance matrix  $K(t, s)$ .



#### 4.5. Practical Application of the Generalized Karhunen - Loève Expansion.

The generalized Karhunen - Loève expansion and its optimalities may find practical applications in designing a suboptimal procedure for feature selection and ordering in pattern recognition systems. In automatic classification systems the good feature observations, which are the most representative and informative, should be selected by a preprocessor prior to initiation of the recognition process. One may choose to minimize the mean square error and select the coordinate system (the generalized Karhunen - Loève system) whose coordinate coefficients represent the pattern samples of different classes in the most significant manner. The minimized entropy property of the Karhunen - Loève expansion implies that the linear transformation produces the most efficient information compression over the coordinate system in the sense that most of the random coefficients are concentrated in a few coordinates instead of widely distributed among all of them.

By properly constructing the generalized Karhunen - Loève coordinate system, and arranging the coordinate functions  $\{\phi_k(t)\}$  (continuous case) or the coordinate vectors  $\{\phi_k\}$  (discrete case) according to descending order of their associated eigenvalues  $\sigma_k^2$ , the resulting feature observations will always contain the maximum information about the input pattern samples whenever the recognition process terminates at a finite number of measurements.



In practice, it is difficult to construct the desired coordinate system through the integral equations. However, one can achieve the same purpose by simply recognizing and applying the necessary and sufficient conditions for the expansion to be the generalized Karhunen - Loève expansion (36).

#### 4.5.1. Necessary and Sufficient Conditions for the Generalized Karhunen - Loève Expansion.

Fu and Chien stated the necessary and sufficient conditions for a generalized Karhunen - Loève expansion to be

$$1) \sum_{i=1}^m P_i E(V_{ik} V_{il}^*) = \sigma_k^2 \delta_{kl},$$

where  $\delta_{kl}$  is the Kronecker delta function,

$$\text{and } 2) \sigma_k^2 = \sum_{i=1}^m P_i \text{Var}(V_{ik}).$$

The proof for sufficiency has been given in sections 4.2.1., 4.3.1. and 4.3.2. during derivation of the optimal properties of the generalized Karhunen - Loève expansion.

The necessity may be proven as follows:

Assume that the covariance function  $K(t, s)$  is defined as

$$K(t, s) = \sum_{k=1}^{\infty} \sigma_k^2 \phi_k(t) \phi_k^*(s)$$

where  $\{\phi_k(t)\}$  are determined by

$$\int_0^T K(t, s) \phi_k(s) ds = \sigma_k^2 \phi_k(t).$$

It may be noted that

$$V_{ik} = \int_0^T X_i(t) \phi_k^*(t) dt \quad [4.24]$$

which is obtained by application of integration to [4.1]





over the range  $(0, T)$ . Similarly,

$$V_{i\ell}^* = \int_0^T X_i^*(s) \phi_\ell(s) ds. \quad [4.25]$$

Substituting [4.24] and [4.25] into condition 1, it follows that

$$\begin{aligned} \sum_{i=1}^m P_i E(V_{ik} V_{i\ell}^*) &= \sum_{i=1}^m P_i E\left[\left\{\int_0^T X_i(t) \phi_k^*(t) dt\right\} \left\{\int_0^T X_i^*(s) \phi_\ell(s) ds\right\}\right] \\ &= \int_0^T dt \phi_k^*(t) \int_0^T ds \sum_{i=1}^m P_i E[X_i(t) X_i^*(s) \phi_\ell(s)] \\ &= \int_0^T dt \phi_k^*(t) \int_0^T ds \sum_{i=1}^m P_i E[X_i(t) X_i^*(s)] \phi_\ell(s) \\ &= \int_0^T dt \phi_k^*(t) \int_0^T ds K(t, s) \phi_\ell(s) \\ &= \int_0^T dt \phi_k^*(t) \sigma_\ell^2 \phi_\ell(t) \\ &= \sigma_k^2 \int_0^T \phi_k(t) \phi_\ell^*(t) dt \\ &= \sigma_k^2 \delta_{k\ell}. \end{aligned}$$

Therefore, construction of the desired coordinate system is equivalent to finding the coordinate function (or vector) in which the coordinate coefficients are mutually uncorrelated so that equations in [4.5] are satisfied. The procedure is basically that of de-correlating the coordinate coefficients over the ensemble of all pattern samples from different classes. In many recognition processes where the covariance functions are real and symmetric, this de-correlation process simply amounts to diagonalization of the corresponding covariance function.

#### 4.6. Procedure for Formulation of the Karhunen - Loève System. (20)

Consider a pattern recognition system consisting of preprocessor and classifier. The preprocessor is designed



to select and order the features by choosing an optimal coordinate system. Let the set of features that describe a pattern sample be denoted by the vector,

$$X_i = (X_{i1}, X_{i2}, \dots, X_{ik})$$

where  $i$  is the index for pattern class,  $i = 1, 2, \dots, m$ , ( $m$  = the total number of classes) and  $k$  is the total number of features observed for each pattern sample. We define feature selection and ordering as the process of deciding upon the proper sequence of feature observations for a particular classifier. The proposed optimal coordinate system (the generalized Karhunen - Loève system) can be determined by application of the following steps which are also summarized in Fig. 15.

- 1) Obtain the covariance function  $K(t, s)$  from the given sample measurement vectors. If the components of the sample vectors assume real values, the covariance function  $K(t, s)$  is a real symmetric matrix.
- 2) Find the eigenvalues, and the associated eigenvectors, of  $K(t, s)$ . Let the eigenvectors be normalized and lexicographically arranged according to descending order of their associated eigenvalues. The set of orthonormal vectors thus obtained constitutes the generalized Karhunen - Loève coordinate system.
- 3) Make the linear transformation as defined in [4.23] where the  $u_{nj}$ 's are the components of the



orthonormal eigenvectors obtained from step 2.

The resulting  $V_{in}$ 's are the desired coordinate coefficients in terms of the generalized Karhunen - Loève coordinate system.

It may be noted that the set of  $V_{in}$ 's is the set of new (or transformed) features to be observed by the classifier. Then for

$$V_i = (V_{i1}, V_{i2}, \dots, V_{ik}),$$

$$V_{in} = \sum_{j=1}^k \chi_{ij} u_{nj}^*, \quad j = 1, 2, \dots, k,$$

$$n = 1, 2, \dots, k,$$

where  $u_{nj}^*$  is the jth component of the nth coordinate vector  $\phi_n$ , in the generalized Karhunen - Loève system  $\{\phi_n\}$ .



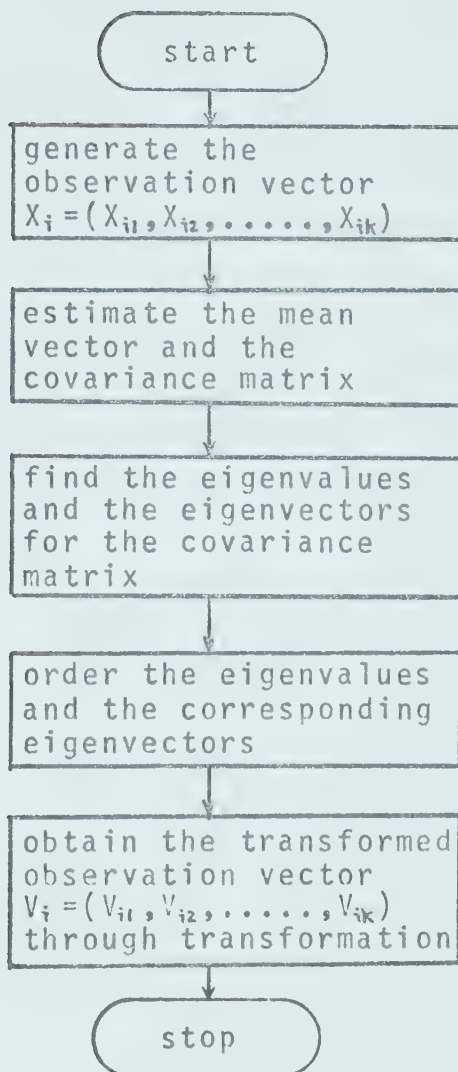


Fig. 15. Flowchart Showing the Procedure for Formulation of the Karhunen - Loève System.





## CHAPTER V

## DATA BASE

5.1. The CACM Data Base.

The data base used to test the proposed automatic document classification system is composed of keywords selected from 179 titles of papers published in CACM, the Communications of the Association for Computing Machinery, during the years 1968, 1969, 1970, and 1971 (Volumes 11 - 14). All these papers were pre-classified when published; they were classified according to the CACM classification schedule. These pre-classified documents can be used as the input test data for the proposed classification system, they provide standard answers to measure the efficiency of the system. Significant keywords, which emphasize the dissimilarities between papers of different classes, are selected from the data base. For the individual classes certain sample statistics, such as the sample mean vector and sample covariance matrix, are extracted and form the required a priori information for the Bayes' classifier.

The CACM data base is punched on cards. An individual datum in the data base consists of keywords selected from the title of a document together with a pre-assigned class number of the document. The pre-assigned class number constitutes the last logical record of each physical record. For each card of 80 columns, the first two columns are reserved for a right justified integer that serves as a header to indicate the number of subsequent logical record fields that contain



information about the document. The remaining 78 columns of the card are divided into 13 logical record fields each of six columns. Left justified truncated keywords of five characters are stored in these logical record fields, and the sixth column of each field is always left blank. Each keyword of less than five characters is padded with appropriate number of blanks on its right hand side. The pre-assigned class number is a digit between 1 and 5, and the number is placed in the second column of the last logical record.

The instance in which the header equals 13 requires special consideration. If the thirteenth logical record field is occupied by a digit in the second column and by blanks elsewhere then the document information occupies 13 logical record fields, and the digit in the second column of the thirteenth logical record field is the pre-assigned class number. On the other hand, if the thirteenth logical record field is occupied by a string of alphanumeric characters with the first character starting from the first column of this field, then the document information occupies more than 13 logical record fields and further information is recorded on the next card.

A listing of the keywords selected from the title of individual paper in the CACM data base can be found in Appendix 1. An example of a single documnet record is the following:



5ANALY TIME SHARI TECHN 1.

It is a document whose pre-assigned class number is 1.

## 5.2. Selection of Keywords.

Three types of features may be measured from a pattern, namely, physical features, topological features, and statistical features. Physical and topological features are commonly found in the recognition process used by human beings. Such features are easily detected by human eyes, by touch, and by other sensory organs. Since computer lacks human sensory organs, the physical and topological features are not the most efficient features in automatic recognition processes. However, the computer may be designed to extract mathematical or statistical features from sample patterns which humans may have difficulty in determining manually. When the patterns in each of the  $m$  pattern classes are random variables governed by  $m$  distinct probability density functions the computer may be taught to perform classifications based on sample statistics.

Computers have memory and are capable of recognizing, comparing, and identifying words. In automatic document classification, the document is often represented by a set of selected keywords. The computer performs recognition based on identification of certain keywords rather than on understanding of their semantic meaning. Words related to the original keywords may be added in order to amplify the information in the stored document representation. Given a set of documents of a data base, the assignment of keywords



for each individual document constitutes the sensing problem.

It is seldom practical to assign keywords by reading manually over the entire document. Not only would such a method involve too much manual work, but the keywords so assigned would necessarily be based on subjective considerations. They would only reflect a particular indexer's interpretation of the contents of the paper. Much manual work may be saved by selecting keywords from an abstract of the paper; however, abstracts are not always present in scientific papers. Also a relatively large amount of processing is still required.

Statistical examinations show that the authors of scientific papers tend to choose the titles of their publications with care; the title of a scientific paper often gives a good indication of the contents of the paper (37). These titles often provide a satisfactory source of appropriate keywords for scientific papers.

Elimination of keywords that are useless in the recognition process is a legitimate strategy in information compression. It has therefore been used in preparation of the data base for the present investigation. In selecting keywords from the title, any insignificant words such as the articles, the prepositions, and the conjunctions, which contribute no discriminatory information in the recognition process, have been ignored. To provide a standard for consistent keyword expression, all the selected keywords were expressed in singular form. Each keyword selected was right-truncated to five characters. This procedure helps to





eliminate the parts of speech problem since words arising from the same stem tend to be coded in the same form. As an example, the words such as computer, computers, computed, computing, computation, computational and computibility are all from the same stem "compute". When truncated to five characters, all these words are coded and stored in the data base as "COMPU". Truncating keywords also has the advantage of saving storage which may well constitute the major cost in manipulation of a large data base.

### 5.3. Selection of Classes.

The Communications of the Association for Computing Machinery uses 13 classes to classify the submitted papers in Computer Science. These 13 classes are as follows:

- 1) Algorithm,
- 2) Computer Systems,
- 3) Education,
- 4) Graphic and Image Processing,
- 5) Information Retrieval,
- 6) Management / Data Base Systems,
- 7) Management Science / Operations Research,
- 8) Numerical Mathematics,
- 9) Operating Systems,
- 10) Programming Languages,
- 11) Programming Techniques,
- 12) Scientific Applications,
- 13) Standards.

Owing to the nature of the subjects involved in class 1



and class 13, the documents in these two classes are difficult for recognition by an automatic document classification system. There was only one paper in the field of "Graphic and Image Processing" published in the CACM between the years of 1968 and 1971; therefore, there is not enough data to form sample statistics for the documents of class 4. For economic reasons, the author used five classes of the remaining ten classes to test the efficiency of the proposed automatic document classification system. The five chosen classes are as follows:

- 1) Computer Systems,
- 2) Information Retrieval,
- 3) Operating Systems,
- 4) Programming Languages,
- 5) Programming Techniques.

#### 5.4. Statistics of the CACM Data Base.

The CACM data base used in the present study is composed of keywords selected from 179 titles of papers published in the Communications of the Association for Computing Machinery during the years 1968, 1969, 1970, and 1971. These keywords are selected from the titles of documents in the five selected classes. The distributions of the sample documents in these classes are summarized in Table I.



Class	Description	No. of Documents	Percentage
1	Computer Systems	21	11.73
2	Information Retrieval	20	11.17
3	Operating Systems	33	18.44
4	Programming Languages	44	24.58
5	Programming Techniques	61	34.08

Table I. Distributions of the Sample Documents in the 5 Classes.

The above statistics show that an average of 36 sample documents are used to represent each class, and the sample documents in class 5 constitute the largest bulk of the data base (34.08%).

A total of 909 selected keywords are used to describe the contents of the 179 sample documents, and this gives an average of 5.1 selected keywords per document.

There are 381 distinct keywords in the CACM data base, therefore each selected keyword occurs on the average of 2.6 times in the entire collection of the sample documents. A listing of the 381 distinct keywords in the CACM data base is shown in Appendix 2.

Fourteen is the maximum number of keywords used to represent a document in the data base, identical keywords in the same title appear only once. A minimum of one keyword is used to represent a document.



## CHAPTER VI

### THE PROPOSED CLASSIFICATION SYSTEM

#### 6.1. Introduction.

The creation of the proposed automatic document classification system can be divided into six phases, namely,

- 1) Feature Preselection Phase,
- 2) Association Measures Assignment phase,
- 3) Complete Fuzzy Relations Assignment Phase,
- 4) Feature Selection and Feature Ordering Phase,
- 5) Sample Statistics Estimation Phase,
- 6) Classification Phase.

The first three phases are designed to determine the strongest possible fuzzy relations between the distinct keywords in the data base; while the feature selection and feature ordering phase serves to provide a compressed data base for the classifier. The sample statistics estimation phase necessarily precedes the classification because the Bayes' classifier is chosen for use in the classification process. This classifier requires the sample statistics, such as the mean feature vectors and the covariance matrices, to serve as the statistical data for classification. Thus the first five phases are designed essentially to prepare a suitable data base for the Bayes' classifier so that an efficient classification system may be obtained.

It may be noted that much computation time is required in the first five phases of the proposed system because they





necessarily have to involve all the distinct keywords in the data base. However, this large amount of computation time is worthwhile in the sense that these phases prepare a compressed data base for the Bayes' classifier; the process of classification is much simplified by the fact that only a few significant features will be observed and they form the bases for classification. Since the first five phases are required only once, in the long run, the computation times that are saved in the classification process will more than compensate for the initial large investment of time.

A detailed description of the creation of the proposed system will be presented in the subsequent sections.

## 6.2. Feature Preselection Phase.

Since the title of each document is used to describe its content, the titles of all the sample documents in the data base were first examined manually by the system designer. This was in order to choose the important words from the titles of the documents. The selected keywords were, in fact, most title words other than the prepositions, articles, and conjunctions. The designer also had to use his own intuition to exclude those title words which obviously would not add to an understanding of the document's content. In the instance that the designer was given a document having the title "Improving Round-off in Runge-Kutta Computation with Gill's Method", he should describe the document with the following eight keywords:

IMPRO ROUND OFF    RUNGE KUTTA COMPU GILL' METHO.



It may be noted that hyphenated title words were treated as two separate words; the selected keywords were all expressed in singular form and were right-truncated to five characters.

The selected keywords of each document together with its pre-assigned class number were punched on card(s), and this string of information was regarded as a physical record of the data base. The collection of cards for all the sample documents constituted the data base for the proposed system.

The procedure used for the feature preselection phase is summarized in Fig. 16.

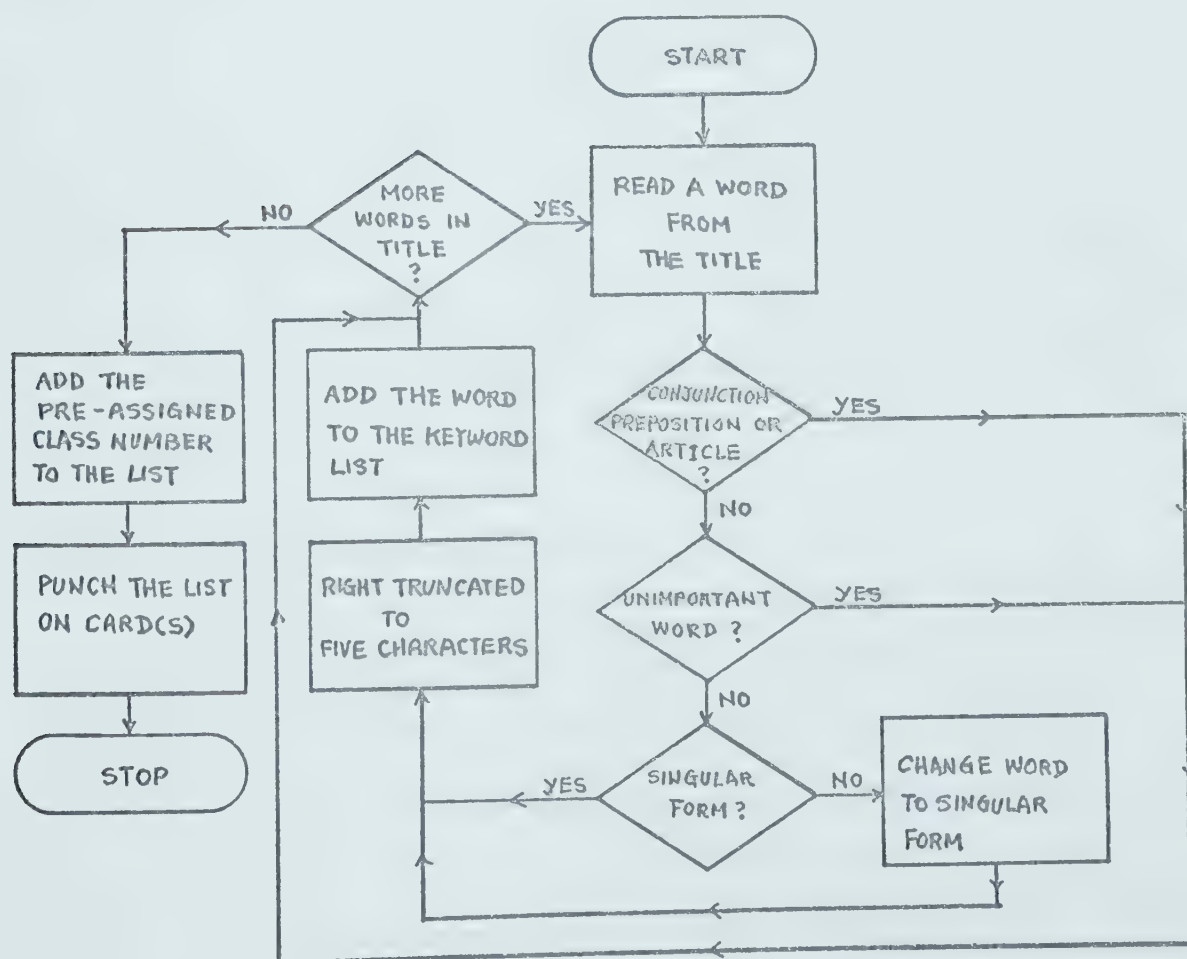


Fig. 16. The Flow Diagram for the Feature Preselection Phase.



### 6.3. Association Measures Assignment Phase.

For each selected keyword of the sample document, it is necessary to assign a vector of values to indicate the associations between the selected keyword and the other distinct keywords in the data base. One way to obtain these values is by finding the statistical association measures between the distinct keywords in the data base.

The strings of information for individual sample documents in the data base were input to the computer. The selected keywords and their corresponding occurrences in terms of document numbers were recorded in logical records. Thus each logical record contains a keyword and its corresponding occurrence. The keywords in the logical records were sorted by the IBM sort and merge package in order to be ranked in the ascending order according to the ASCII code (the ASCII code places the special characters in front, with the alphabetical characters in the middle, and the integers at the end).

#### 6.3.1. Document Term Matrix.

The sorted logical records were input to the computer. The distinct keywords were extracted from the sorted keyword list. At the same time a document term matrix, with the distinct keywords along one axis and the document numbers along the other, was formed. This matrix indicated the keywords that would be found in a particular document. Those distinct keywords of the data base that did not appear in the title of a particular document were tagged with the logical



constant "false", and the distinct keywords that were present in the document were tagged with the logical constant "true". This arrangement allowed some saving in storage because a logical\*1 matrix could be used to store the required information.

#### 6.3.2. Term Connection Matrix.

The number of times of co-occurrence of a particular pair of distinct keywords in the sample documents of the data base can be measured by linking the keywords through their corresponding occurrences. A term connection matrix was formed by multiplying the document term matrix with its transpose so that each element of the resulting term connection matrix indicated the number of documents that were relevant to that pair of keywords. The term connection matrix also gave some indication of the relations between the distinct keywords in the data base. Keywords that often appear together in the same documents may be regarded as having a close relationship.

#### 6.3.3. Term Relation Matrix.

The term relation matrix may be regarded as the normalized version of a term connection matrix such that each element of the term relation matrix satisfies the fuzzy set property. In other words, the elements of the term relation matrix are all governed by a membership function and all their values lie in the closed interval between zero and one. A statistical association measure formula proposed





by L. B. Doyle (12) was used to perform the required conversion. The association measure formula may be expressed in the following form:

$$\text{TRM}(i, j) = \frac{\text{TCM}(i, j)}{\text{TCM}(i, i) + \text{TCM}(j, j) - \text{TCM}(i, j)}$$

where  $\text{TRM}(i, j)$  is the  $i$ th row and the  $j$ th column element of the term relation matrix which indicates the similarity relation between the  $i$ th and  $j$ th keywords,

$\text{TCM}(i, j)$  is the  $i$ th row and the  $j$ th column element of the term connection matrix which indicates the number of documents relevant to both the  $i$ th and  $j$ th keywords,

$\text{TCM}(i, i)$  is the  $i$ th diagonal element of the term connection matrix which indicates the number of documents relevant to the  $i$ th keyword,

and  $\text{TCM}(j, j)$  is the  $j$ th diagonal element of the term connection matrix which indicates the number of documents relevant to the  $j$ th keyword.

The expression  $\text{TCM}(i, i) + \text{TCM}(j, j) - \text{TCM}(i, j)$  in the denominator represents the number of documents that contain either, but not both, of the  $i$ th and  $j$ th keywords. It has a normalizing effect and ensures that the value of every element in the term relation matrix will lie in the closed interval between zero and one, and thus satisfy the fuzzy set property.

It may be noted that the term relation matrix is equivalent to a 1 - step fuzzy relation matrix because the calculated values of  $\text{TRM}(i, j)$ 's indicate the direct



relations between the  $i$ th and  $j$ th keywords, and all these values lie in the closed interval between zero and one.

A summary of the association measures assignment phase is shown in Fig. 17.

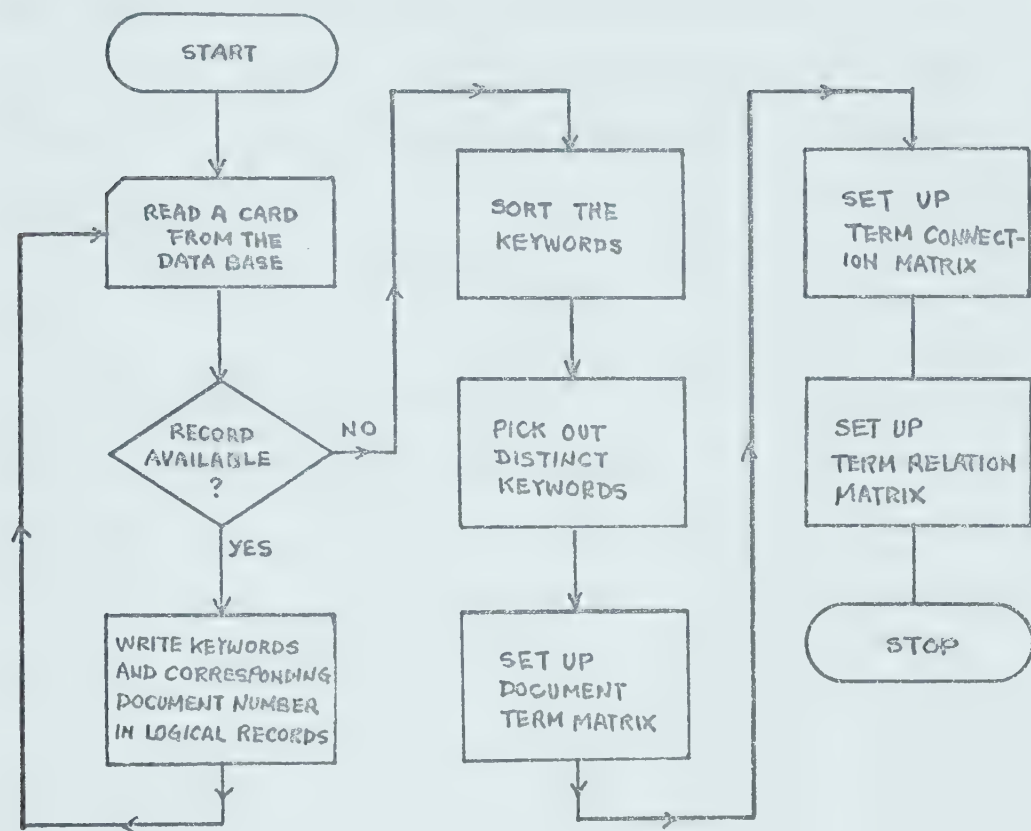


Fig. 17. The Flow Diagram for the Association Measures Assignment Phase.

#### 6.4. Complete Fuzzy Relations Assignment Phase.

Many distinct keywords in the data base did not have direct relations with each other; therefore a considerable number of zeroes were present in the 1 - step fuzzy relation matrix. In order to extract the maximum possible relations between keywords in the data base, the indirect relations between keywords should also be used. An indirect relation



between a pair of keywords may be defined as the relation obtained when the keywords are linked together by a chain of  $n$  keywords that connects them. In the instance when one keyword is used as the connection between a pair of keywords the relation measure so obtained is known as the 2 - step relation. The 2 - step fuzzy relation matrix may be obtained from the 1 - step fuzzy relation matrix by applying the following max - min composition operation in fuzzy logic (39):

$$\mu_{A \circ A}(x, y) = \max_v \min[\mu_A(x, v), \mu_A(v, y)]$$

$$v = 1, 2, \dots, n.$$

Consider, for example, a 1 - step fuzzy relation matrix with four distinct keywords whose direct relations are shown in Fig. 18.

	K(1)	K(2)	K(3)	K(4)
K(1)	1.0	0.4	0.1	0.3
K(2)	0.4	1.0	0	0.2
K(3)	0.1	0	1.0	0.7
K(4)	0.3	0.2	0.7	1.0

Fig. 18. The 1 - step Fuzzy Relation Matrix.

From the 1 - step fuzzy relation matrix, it may be noted that there is no direct relation between keyword 2 and keyword 3. However, direct relations do exist between keywords 1 and 2 and keywords 1 and 3, with the resulting values equal to 0.4 and 0.1 respectively. Likewise, there are relations between keywords 4 and 2 and keywords 4 and 3 with the values equal to 0.2 and 0.7 respectively. By applying the max - min composition operation on the 1 - step



fuzzy relation matrix, one should be able to obtain the indirect 2 - step fuzzy relation between keywords 2 and 3 through keyword 4 with value equals to 0.2.

Based on the fact that  $A^{2n} = A^n \circ A^n$ , we can obtain the 2n - step from the n - step fuzzy relation matrix rather quickly and efficiently. Certainly, in order to obtain the complete fuzzy relations between the distinct keywords in the data base, we have to apply the max - min composition operation repetitively until the elements in the 2n - step fuzzy relation matrix are exactly the same as those in the n - step case. However, for economic reasons, the process may be discontinued as soon as most of the elements remain relatively stable, and the last 2n - step fuzzy relation matrix may be used as the complete fuzzy relation matrix for the distinct keywords in the data base.

In our experiment, the elements remained relatively stable at the 32 - step, therefore, the 32 - step fuzzy relation matrix was taken as the complete fuzzy relation matrix.

A flow diagram showing the procedure for finding the complete fuzzy relation between keywords in the data base is shown in Fig. 19.





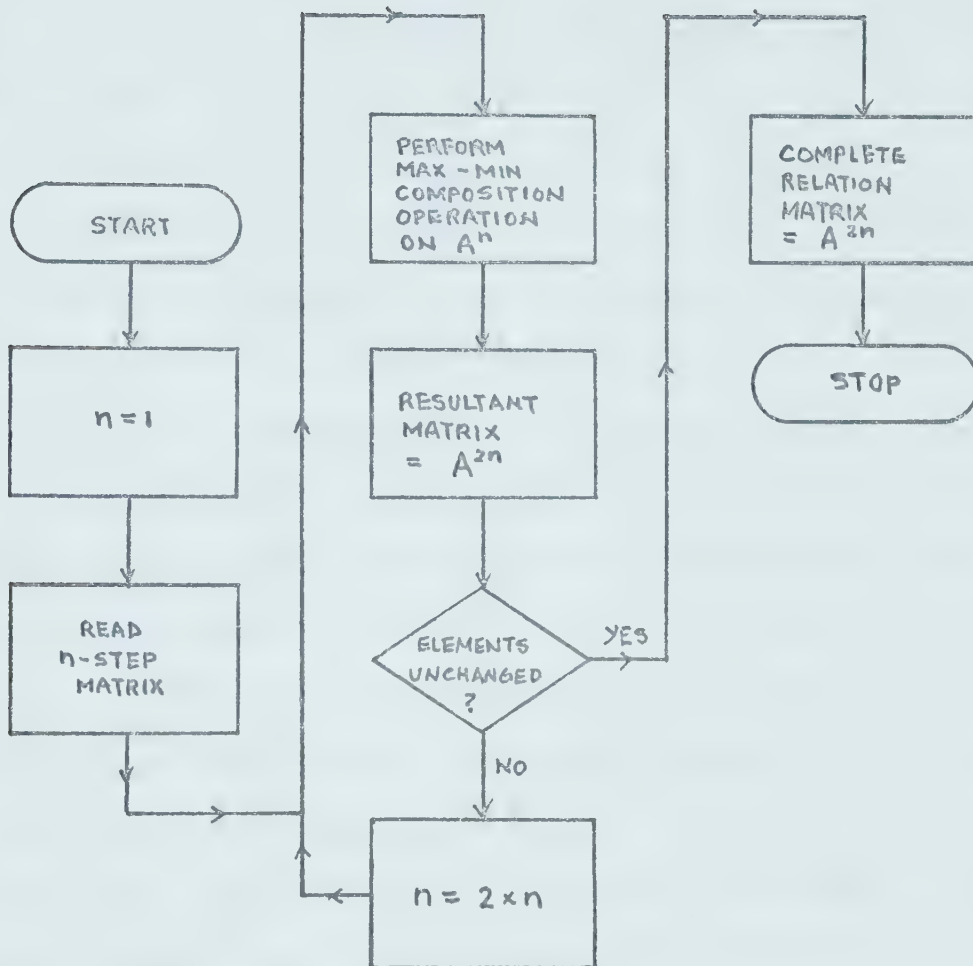


Fig. 19. The Flow Diagram for the Complete Fuzzy Relations Assignment Phase.

#### 6.5. Feature Selection and Feature Ordering Phase.

Of the four feature selection methods described in Chapter II, the Karhunen - Loève expansion method seems to be most appropriate for the proposed automatic document classification system. The application of the Karhunen - Loève expansion in feature selection requires determination of the distinct keywords in the data base.



### 6.5.1. Feature Vector.

The sample documents in the data base were input to the computer. The selected keywords and their corresponding occurrences in terms of document numbers were recorded in logical records. These records were sorted by the IBM sort and merge package so that the keywords in the logical records were ranked according to the ASCII code. Each sorted keyword was compared with the keywords in the distinct keyword list. If there was a match between the sorted keyword and a keyword in the distinct keyword list, the corresponding document number of the sorted keyword was recorded. The relation vector between the matched keyword and the other distinct keywords in the data base was extracted from the complete fuzzy relation matrix; this relation vector forms part of the constituents of the feature vector for that particular document.

Consider, as an example, a document described by three keywords. Three relation vectors would be extracted from the complete fuzzy relation matrix. By summing these three vectors, element by element, the feature vector for that particular document is obtained.

### 6.5.2. Mean Feature Vector.

In our experiment there were 381 distinct keywords and 179 documents in the data base. As a consequence, there were 179 feature vectors each having 381 elements. Using these feature vectors to form the rows of a matrix, the resultant matrix was the feature matrix for the sample documents in the



data base with 179 x 381 elements. The mean feature vector was obtained by summing the elements in the matrix columnwise and dividing the resulting vector by 179 (since there were 179 documents in the data base).

#### 6.5.3. The Covariance Matrix.

The covariance matrix can be obtained by applying the statistical formula in the form:

$$\Sigma = \frac{1}{n} \sum_{i=1}^n [(X_i - \bar{X})(X_i - \bar{X})']$$

where  $\Sigma$  is the covariance matrix of the data base,

$n$  is the number of documents in the data base,

$X_i$  is the feature vector for the  $i$ th document in the data base,

and  $\bar{X}$  is the mean feature vector of all the documents in the data base.

In our experiment, the covariance matrix was a 381 x 381 matrix which described the correlations between the distinct keywords in the data base.

#### 6.5.4. The Transformation Matrix.

The next step is to find the transformation matrix for the Karhunen - Loève expansion. The transformation matrix is formed by the selected eigenvectors of the covariance matrix. In our experiment, the dimension of the covariance matrix was rather large, it was convenient to use the IMSL subroutine packages to find the required eigenvectors. The original covariance matrix was transformed into a tridiagonal matrix by the Householder reduction method. In



other words, the original matrix was transformed into a matrix having the information concentrated on the elements along the main diagonal and the two subdiagonals of the matrix. It may be noted that the details about this transformation must be saved so that the eigenvectors of the original matrix may be restored in the subsequent stage. The Q. L. algorithm which evolved from the Q. R. algorithm was used to find the eigenvalues and the eigenvectors of the tridiagonal matrix. To obtain the eigenvectors of the original covariance matrix, we had to make use of the transformation details recorded previously.

#### 6.5.5. Feature Ordering.

The eigenvectors and their corresponding eigenvalues were recorded in logical records. The IBM sort and merge package was used to arrange the eigenvalues in descending order. The eigenvectors corresponding to the top twenty eigenvalues were used to form the required transformation matrix. In our experiment, the transformation matrix was a 20 x 381 matrix with its 20 rows formed by the 20 selected eigenvectors.

#### 6.5.6. The Compressed Data Base.

According to the theory of the Karhunen - Loève expansion, the compressed data base can be obtained by multiplication of the original matrix with the transformation matrix. The compressed relation matrix in our experiment was a 20 x 381 matrix with the 20 most significant keywords





along one axis and the 381 distinct keywords in the data base along the other. This matrix revealed the complete relations between the 20 significant keywords and the 381 distinct keywords in the data base. A summary of the feature selection and feature ordering phase is shown in Fig. 20.

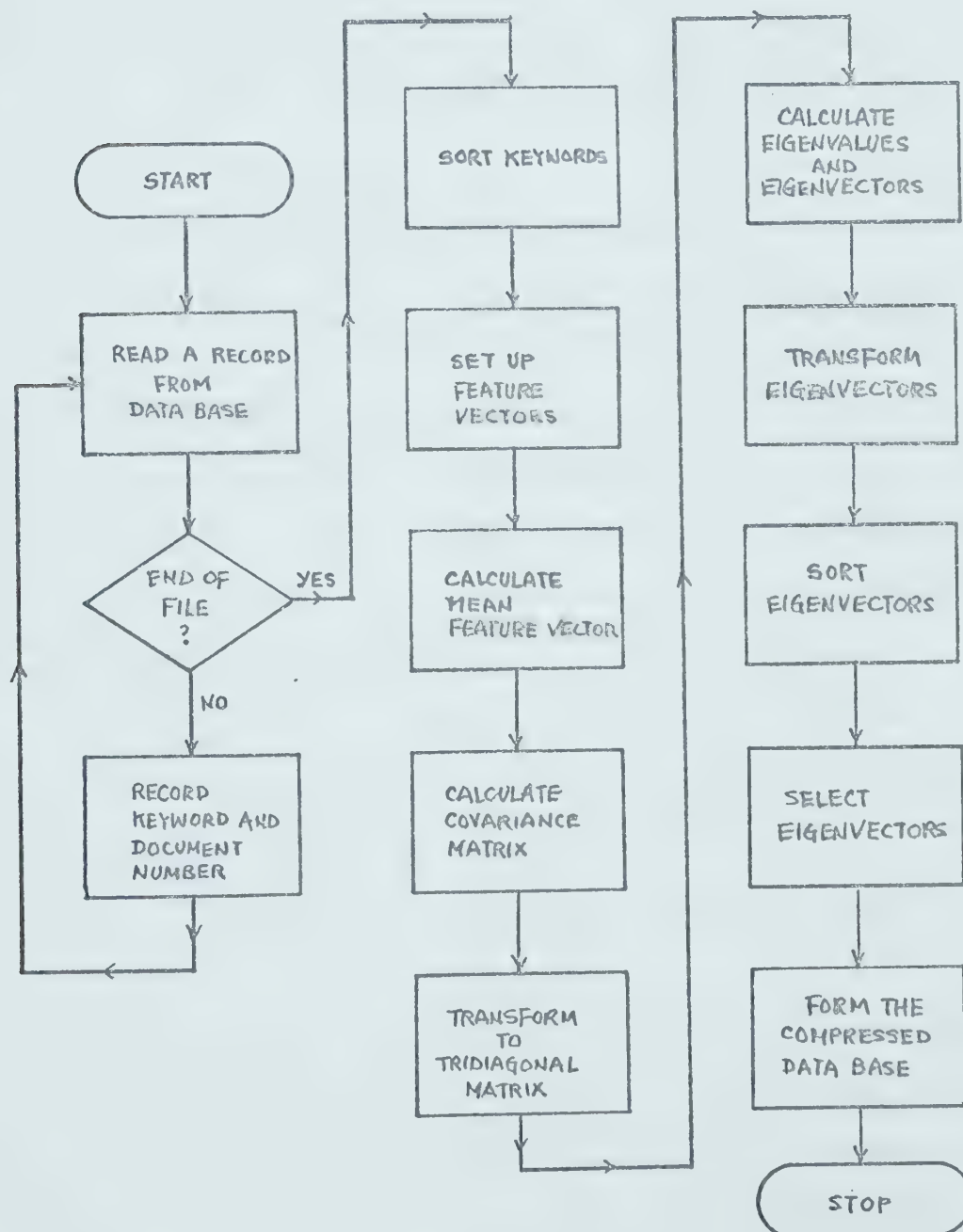


Fig. 20. The Flow Diagram for the Feature Selection and Feature Ordering Phase.



### 6.6. Sample Statistics Estimation Phase.

The sample documents in the data base were sorted into 5 classes according to the pre-assigned class numbers. Documents from a particular class were input to the computer, and the selected keywords of each document were compared with the distinct keywords in the data base. If a match occurred, the relation vector between the selected keyword and the 20 significant keywords was extracted from the compressed relation matrix. The same procedure was repeated for all other selected keywords of the document. Summation of these relation vectors gave the feature vector for the document with respect to the 20 significant keywords. The mean feature vector for all the documents in that class was obtained by summing the feature vectors and dividing the resultant vector by the number of documents in that class. The covariance matrix for a particular class of documents was obtained by the following statistical formula:

$$\Sigma_i = \frac{1}{n_i} \sum_{j=1}^{n_i} [(X_{ij} - \bar{X}_i)(X_{ij} - \bar{X}_i)'], \quad i = 1, 2, 3, 4, 5,$$

where  $\Sigma_i$  is the covariance matrix for class  $i$  documents,

$n_i$  is the number of documents in class  $i$ ,

$X_{ij}$  is the feature vector for the  $j$ th document in class  $i$ ,

and  $\bar{X}_i$  is the mean feature vector for class  $i$  document.

In our experiment, the covariance matrix for each class was a 20 x 20 real symmetric matrix. The same sample statistics finding routine was used to obtain the mean feature vector and covariance matrix for other classes.



These sample statistics for the five selected classes form the required a priori information for the Bayes' classifier.

### 6.7. Classification Phase.

According to the theory of the parametric training method stated in Chapter II, the Bayes' classifier using the maximum likelihood discriminant rule for five classes may be expressed in terms of five discriminant functions  $G_i(X)$ ,

$i = 1, 2, \dots, 5$ , each of the form (29):

$$G_i(X) = b_i - \frac{1}{2}[(X_i - \bar{X}_i)' \Sigma_i^{-1} (X_i - \bar{X}_i)]$$

where  $b_i = \log_e P(W_i) - \frac{1}{2} \log_e |\Sigma_i|$ .

$|\Sigma_i|$  is the determinant of the covariance matrix for class  $i$  documents,

$\Sigma_i^{-1}$  is the inverse of the covariance matrix for class  $i$  documents,

$X_i$  is the feature vector of the document to be classified,

and  $\bar{X}_i$  is the mean feature vector for class  $i$  documents.

In the expression for  $b_i$ ,  $P(W_i)$  is the a priori probability of occurrence of class  $i$  documents; it may be calculated by the formula:

$$P(W_i) = \frac{Z_i}{\sum_{i=1}^5 Z_i}$$

where  $Z_i$  is the number of occurrences of class  $i$  documents in the data base.

The five discriminant functions produced five scalar



values each indicating the likelihood that the document should be classified in that particular class. The class number which corresponded to the maximum scalar value of the  $G_i(X)$ 's was assigned to the document as indicating the class to which it should belong.

A listing of the classification program can be found in Appendix 3.





## CHAPTER VII

### RESULTS AND STATISTICS

#### 7.1. Test Data.

The test data was made up of 125 documents that were randomly selected from the Communications of the Association for Computing Machinery between the years of 1968 and 1972. These documents had each been pre-classified by the CACM into one of the five selected classes, and they thus provided the necessary standard answers for the tests. The test data was divided into five groups, each consisting of 25 documents. The constituents of the first four groups were those documents published in the CACM between the years of 1968 and 1971; these documents were also part of the sample documents that made up the statistical data of the proposed classification system. In order to test for the performance of the system on documents other than those in the data base, the fifth group included in the data base was formed from documents published in the year 1972. The proposed document classification system was applied to documents of each group in turn and assigned each document to an appropriate class. It may be noted that a "two ranks classification system" was used in the classifications; each document was first assigned to its most appropriate class (the first rank classification) and then assigned to its next appropriate class (the second rank classification). This system has the advantage of increasing the percentage of recall, and the second rank classification usually proves to be very useful



in real life classification problems.

## 7.2. Programming Details.

Initially, all the papers published in the Communications of the Association for Computing Machinery between the years of 1965 and 1971 (7 years) were used as the sample documents in the data base. This accounts for a total of 307 documents with 4055 keywords and 1008 distinct keywords. The keywords were selected from both the titles and the abstracts of these papers. However, when constructing the term relation matrix, it was found that the program required at least four times one million bytes for storage (the term relation matrix is a  $1008 \times 1008$  real valued matrix). Since the computer centre at the University of Alberta offers a maximum of one million bytes of core memory there was not enough memory space to manipulate such a large matrix in core. For economic reasons, the author was forced to trim the data base and included only those papers from the five selected classes published between the years of 1968 and 1971.

It may be noted that minimizing the storage space is an important factor that merits special attention in large data base handling. Since there are 381 distinct keywords in the experimental data base, the programs for the first four phases as described in Chapter VI have to handle a square matrix of  $381 \times 381$  elements at all times. It is obvious that a  $381 \times 381$  logical\*1 matrix may be used to store the elements of the document term matrix; this saves three quarters of the usual storage requirement. The term



connection matrix is an integer valued matrix, and the value of each element can be stored in two bytes in a integer\*2 matrix. As for the term relation matrix and the n - step term relation matrix, a full word is necessary to store the value of each element. By making use of the symmetric property of these matrices, only the upper triangle of the matrix has to be calculated; the lower triangle is merely the mirror image of the upper half.

In the feature selection and ordering phase, the CS002A subroutine in the University of Alberta Computing Science Program Library (39) was used to find the eigenvalues and eigenvectors of the covariance matrix. The routine employs full storage mode. Householder's method of tridiagonalizing the input symmetric matrix is used with a variant of the Q. R. algorithm to find the eigenvalues. The eigenvectors are found using inverse iteration. However, this subroutine program is not suitable for a huge matrix. In one particular run, the program used 4 minutes CPU time with an elapsed time of 6 hours and 1,000,000 drum reads; the program was stopped by the computer operator. It is obvious that a paging problem exists in this subroutine, and using full storage mode to store a huge symmetric matrix may well be the main cause of this paging problem. The author was advised to use the International Mathematical and Statistical Library (IMSL) subroutine programs to tackle the problem. Instead of full storage mode, the symmetric storage mode was used to store the symmetric matrix; only the



lower triangle of the symmetric matrix was input into the computer memory and these elements were stored in a vector form.

The three IMSL subroutines used were EHOUSS, EQRT2S and EHOBKS (40). The EHOUSS routine computes a Householder's reduction of the input real symmetric matrix to a symmetric tridiagonal matrix. It is a modification of the Martin, Reinsch, Wilkinson Algol Procedure TRED3. The EQRT2S routine is designed to find all eigenvalues and eigenvectors of a symmetric tridiagonal matrix. It performs a Q. L. algorithm which is derived from the Q. R. algorithm. The routine is a modification of the Bowdler, Martin, Reinsch and Wilkinson Algol Procedure TQL2. The EHOBKS routine performs a back transformation to derive eigenvectors of the original matrix. It makes use of the details of the transformation in the original matrix which were computed in EHOUSS. The routine is a modification of Martin, Reinsch, Wilkinson Algol Procedure TRBAK3. Although much computer time was still required in the computation, the IMSL routines proved to have solved the paging problem.

The sample statistics estimation phase and classification phase involved only ordinary programming techniques and there are no details worth mentioning.

### 7.3. Experimental Results.

The proposed automatic document classification system gave a fairly high degree of accuracy in document classification. Of the 25 documents in each group, the two





ranks classification scheme correctly assigned all 25 documents in group 1, 23 out of 25 documents in group 2, 21 out of 25 documents in group 3, 22 out of 25 documents in group 4, and 20 out of 25 documents in group 5; these results gave the classification accuracy of 100%, 92%, 84%, 88% and 80% respectively for the five groups. These results are based on regarding a classification as correct if the correct class appears in either the first or the second rank. The author claims that the percentage of correct classifications in the first or second rank for the proposed system is in the order of  $80 \pm 5\%$ . The results of the tests are summarized in Table II.

Statistics showed that the system took approximately 23 seconds of CPU time to classify 25 documents (the figure included the compile time) and this gave an average of 0.92 second to classify each document. It may be of interest to note that an average of \$3.10 was necessary to run a classification program that classified 25 documents and this gave an average cost of 12 cents to classify each document.

A complete printout of the experimental classification results can be found in Appendix 4.

#### 7.4. Discussions and Suggestions.

The high degree of accuracy of the proposed system suggests that automatic classification based on the concept of fuzzy sets is a feasible alternative to manual classification scheme. The proposed system is also superior in terms of time and expense. From the results, it is not surprising to find that the first four groups gave a better classification performance than the last group; this can be



	Group 1	Group 2	Group 3	Group 4	Group 5
Number of Documents in Class	25	25	25	25	25
Number of Correct Classifications Listed in First Rank	23	21	18	20	15
Number of Correct Classifications Listed in Second Rank	2	2	3	2	5
Total Number of Correct Classifications in First and Second Rank	25	23	21	22	20
Percentage of Correct Classifications in First and Second Rank	100%	92%	84%	88%	80%

Table II Experimental Classification Results.



explained by the fact that the documents in the first four groups were also those documents used to calculate the statistical data of the system. However, the sudden drop of number of correct classifications listed in the first rank in group 5 (an average of 21 correct classifications listed in the first rank for the first four groups; 15 correct classifications listed in the first rank for group 5) suggests that the statistical data of the system is not stable enough; more sample documents should be added to the original data base in order to obtain more accurate statistics.

It is believed that the performance of the proposed system may also be improved by some other means. Increasing the number of keywords to represent each document by including keywords extracted from the abstract may well form one such means. However, such an increase will certainly increase the storage space required in the data base preparation, and at the same time will require more manual work in keyword selection at the initial stage. A more fruitful improvement can be achieved by using more significant features in the classification phase; the Karhunen - Loève expansion scheme in feature selection and ordering guarantees a better performance of the system with each increment in the number of significant features.



## CHAPTER VIII

## CONCLUSIONS

This thesis has demonstrated that the concept of fuzzy relations and use of the Karhunen - Loève expansion allows formulation of a feasible statistical approach to the decision making problem that occurs in automatic document classification. The concept of the proposed automatic document classification system is based entirely on the statistical relationships between keywords and subject categories. Keywords of a document are extended by using keyword association. Assuming that the distribution of the feature vector selected is multivariate normal for each pattern class, the mean feature vector and covariance matrix for each class are computed from the training samples. These statistical data are used as the bases for Bayes' classification in the experiment.

Despite the fact that only 20 out of the possible 381 distinct keywords were used as the selected features in the classification, the system performed adequately in the experiments giving  $80 \pm 5\%$  accuracy in document classification. The relatively small number of features being consider in classification does not only save a lot of storage space in preparing the compressed data base; it also saved a considerable amount of computation time would have otherwise been required.

Besides giving fast, economical, and dependable service, the system also offers other desirable features.





It has the advantage of being highly flexible; a higher percentage of classification accuracy is guaranteed when using more selected features in the classification.

No doubt, every system is bound to have its own defects; the proposed system is of no exception. Large investment in terms of CPU time has to be spent in the preparation of the data base, especially during the Complete Fuzzy Relations Assignment Phase. In fact, this phase accounts more than four fifths of the total CPU time used in this project.

It is desirable to have a general classification system which can classify documents from all disciplines. However, because of the computer memory limitation and the nature of the proposed scheme, the system had to limit its classification ability to five specific fields in Computer Science. One may question the feasibility of the proposed system when applied in a large information centre because such application would involve too many keywords and would require computation of a large number of discriminant functions.

Updating constitutes the most serious problem in the proposed system. When there are new developments in the fields of interest there are likely to occur a number of important new keywords which should be added to the original data base. This requires a serious modification of the entire data base and the most tedious portion of the calculations have to be repeated.

Of course, strictly speaking, the proposed automatic document classification system is not totally automatic



since the Feature Preselection Phase is performed manually. It has been suggested that any manual work is undesirable in document classification since it may introduce biased interpretation and hinder the accuracy of the system.

Further developments of the proposed system may include using more sample documents in preparation of the data base so as to enlarge the vocabulary of the system. We expect that an increment in the number of sample documents for the individual classes should give a more stable set of statistical data for each class and hence lead to more accurate classification.

Since the fuzzy relation matrix of the distinct keywords in the data base is symmetric, it is possible to save approximately half the storage space by storing the matrix in symmetric storage mode. This moderation does not only save storage but also eliminates the paging problem which is very common in large data base handling.

To eliminate all manual work in the system, it is suggested that a "stop list" should be used to choose the important keywords in a title. Undesirable keywords are recorded in the stop list, and only those words other than those listed in the stop list would be used to describe the document.

In conclusion, the author believes that automatic computer classification of documents by the method described in the present thesis is a feasible alternative to manual classification because the results have proved that the performance of the former is as good as, if not better than,



the classical manual approach.

In order to make the computer classification more economical, and hence to allow application to larger sets of data, consideration should be given to development of approximate or iterative techniques for manipulation of the large matrices. Investigation of such techniques was believed to be beyond the scope of the present thesis since it was felt necessary to first study the unmodified use of fuzzy relations and the Karhunen - Loève expansion. It is hoped, however, that the present study may be continued with an emphasis on methods of reduction of processing time.



## REFERENCES

1. COOK, G. A. and HEAPS, D. M., "An Experiment in Information Retrieval", Chemistry in Canada, Ottawa, Ontario, Jan., 1970.
2. FOSKETT, A. C., "The Subject Approach to Information", Clive Bingley, London, March, 1970.
3. AKIYAMA, S., "Automatic Document Classification Systems", Master Thesis, Dept. of Computing Science, University of Alberta, Edmonton, Alta., May, 1972.
4. LUHN, H. P., "A Statistical Approach to Mechanized Encoding and Searching of Library Information", IBM Journal of Research and Development, Vol. 1, 1957, PP. 309 - 317.
5. LUHN, H. P., "Statistical Approach to Content Analysis of Document", CACM.
6. LUHN, H. P., "The Automatic Creation of Literature Abstracts", IBM Journal of Research and Development, Vol. 2, 1958, PP. 159 - 165.
7. MARON, M. E. and KUHN, J. L., "On Relevance Probabilistic Indexing and Information Retrieval", JACM, Vol. 7, 1960, PP. 216 - 244.





8. SALTON, G., "Document Retrieval System for Man - Machine Interaction", The ACM Proceedings of the 19<sup>th</sup> National Conference, 1964.
9. SALTON, G., "Automatic Information Organization and Retrieval", McGraw-Hill Series in Computer Science, 1968.
10. MARON, M. E., "Automatic Indexing: An Experimental Inquiry", JACM, Vol. 8, 1961, PP. 404 - 417.
11. STILES, H. E., "The Association Factor in Information Retrieval", JACM, Vol. 8, 1961, PP. 271 - 279.
12. DOYLE, L. B., "Indexing and Abstracting by Association", Amer. Doc. 13, #4, Oct., 1962, PP. 378 - 390.
13. BAKER, F. B., "Information Retrieval Based on Latent Class Analysis", JACM, Vol. 9, Oct., 1962, PP. 512 - 521.
14. LAZARSFELD, P. E. and HENRY, N. W., "Latent Structure Analysis", Houghton Mifflin Company, 1968.
15. SALTON, G., "Association Document Retrieval Techniques Using Bibliographic Information", JACM, Vol. 10, 1963, PP. 440 - 457.
16. GARFIELD, E., "Primordial Concepts, Citation Indexing, and Historio-bibliography", Journal of Library History, 2(3), 1967, PP. 235 - 249.



17. BORKO, H. and BERNICK, M. D., "Automatic Document Classification", JACM, 10(2), April, 1963, PP. 151 - 162.
18. BORKO, H. and BERNICK, M. D., "Automatic Document Classification Part II, Additional Experiments", JACM, Vol. 11, 1964, P. 138.
19. TAMURA, S., HIGUCHI, S. and TANAKA, K., "Pattern Classification Based on Fuzzy Relations", IEEE Trans. on Systems, Man and Cybernetics, Vol. SMC-1, #1, 1971, PP. 61 - 66.
20. FU, K. S., "Sequential Methods in Pattern Recognition and Machine Learning", Academic Press, New York, 1968.
21. TOU, J. L., "Engineering Principles of Pattern Recognition" in "Advances in Information System Science Vol. 1", Plenum Press, New York, 1969.
22. MARILL, T. and GREEN, D. M., "On the Effectiveness of Receptors in Recognition System", IEEE Trans. on Inform. Theory 9, 1963, PP. 11 - 17.
23. ANDERSON, T. W., "Introduction to Multivariate Statistical Analysis", John Wiley & Sons Inc., New York, 1958.



24. FU, K. S., "Statistical Pattern Recognition" in  
"Adaptive, Learning, and Pattern Recognition  
Systems: Theory and Applications", edited by  
Mendel and Fu, Academic Press, New York, 1970.
25. LEWIS, P. M., "The Characteristic Selection Problem in  
Recognition Systems", IRE Trans. on Inform.  
Theory, Feb., 1962, PP. 171 - 178.
26. MIN, P. J., "A Non-parametric Method for Feature  
Selction", IEEE Proc. 7th Symposium on  
Adaptive Processes, U.C.L.A., Dec., 1968.
27. WATANABE, S., "Karhunen - Loève Expansion and Factor  
Analysis - Theoretical Remarks and  
Application", Proc. 4th Prague Conference  
Information Theory.
28. CHIEN, Y. T. and FU, K. S., "On the Generalized  
Karhunen - Loève Expansion", IEEE Trans. on  
Inform. Theory, IT-13, 1967, PP. 518 - 520.
29. FU, K. S. and LI, T. J., "Formulation of Learning  
Automata and Automata Games", Information  
Sciences, Vol. 1, 1969, PP. 237 - 256.
30. NILSSON, N. L., "Learning Machines, Foundation of  
Trainable Pattern Classification Systems",  
McGraw-Hill Inc., 1965.



31. ZADEH, L. A., "Fuzzy Sets", Information and Control 8, 1965, PP. 338 - 353.
32. ZADEH, L. A., "Shadows of Fuzzy Sets", Problems of Information Transmission, March, 1966, PP. 37 - 44.
33. ZADEH, L. A., "Comm. Fuzzy Algorithm", Information and Control 12, 1968, PP. 94 - 102.
34. ZADEH, L. A., "Similarity Relations and Fuzzy Ordering", Information Sciences, Vol. 3#2, April, 1971, PP. 177 - 200.
35. WATANABE, S., LAMBERT, P. F., KULIKOWSKI, C. A., BUXTON, J. L. and WALKER, R., "Evaluation of Selection of Variables in Pattern Recognition" in "Computer and Information Sciences II" edited by Tou, J. T., Academic Press, New York, 1967.
36. CHIEN, Y. T. AND FU, K. S., "Selection and Ordering of Feature Observations in Pattern Recognition System", Information Control, Vol. 12, May, 1968, PP. 394 - 414.
37. HEAPS, H. S., "A Theory for Automatic Document Classification", Information and Control, in press, 1973.





38. WEE, W. G. and FU, K. S., "A Formulation of Fuzzy Automata and its Application as a Model of Learning Syatems", IEEE Trans. on Systems Science and Cybernetics, Vol. SSC-5, #3, July, 1969.
39. RINES, L. L., "Eigenvalues and Eigenvectors of Real Symmetric Matrices: CS002A", Program Library - IBM 360/67, Library Abst. - Lina, Computing Services, The University of Alberta, Nov., 1969.
40. "INSL Library 1 Reference Manual", 1st Edition, International Mathematical and Statistical Library, Inc., August, 1971.



## APPENDIX 1

Keywords Selected from the Titles of Individual  
Documents in the CACM Data Base.

PP. 114 - 117.







3	PAGIN	ANOMA	3							
4	MEASU	SEGME	SIZE	3						
5	CONCU	CONTR	READE	WRITE	3					
6	PROCE	ALLOC	METHO	TIME	SHARI	3				
8	ANOMA	SPACE	TIME	CHARA	PROGR	PAGIN	MACHI	3		
4	PREVE	SYSTE	DEADL	3						
6	MULTI	MACHI	CODIN	COMPU	ORGAN	3				
5	LOADE	STAND	OVERL	PROGR	3					
6	PERFO	MONIT	TIME	SHARI	SYSTE	3				
9	PROCE	MANAG	RESOU	SHARI	MULTI	ACCES	SYSTE	ESPE	3	
7	INTER	BASED	ORGAN	MANAG	INFOR	SYSTE	3			
7	ORGAN	MATRI	OPERA	PAGE	MEMOR	SYSTE	3			
2	PSEUD	3								
6	DYNAM	SPACE	SHARI	COMPU	SYSTE	3				
4	CONVE	ACCES	MACHI	3						
3	ANOMA	PAGIN	3							
4	SORTI	PAGIN	ENVIR	3						
3	INSTR	MULTI	3							
6	WORKI	SET	MODEL	PROGR	BEHAV	3				
4	PREVE	SYSTE	DEADL	3						
7	USER	PROGR	MEASU	TIME	SHARE	ENVIR	3			
5	MEANI	NAME	PROGR	SYSTE	3					
7	POLIC	DRIVE	SCHED	TIME	SHARI	SYSTE	3			
5	INTER	COMMA	GENER	FACIL	3					
5	GPL	GENER	PURPO	LANGU	4					
8	BOOLE	MATRI	METHO	DETEC	SIMPL	PRECE	GRAMM	4		
6	IMPLE	BASIC	SYSTE	MULTI	ENVIR	4				
3	LRLTR	COMPI	4							
7	COMPL	INHER	AMBIG	CONTE	FREE	LANGU	4			
4	PROOF	PROGR	FIND	4						
4	AUTOM	PROGR	SYNTH	4						
4	RECUR	INDUC	PRINC	4						
4	SIMPL	LR(K)	GRAMM	4						
7	LANGU	EXTEN	GRAPH	PROCE	FORMA	SEMAN	4			
3	DATA	STRUC	4							
5	COMPO	SEMAN	ALGOL	68	4					
5	BLISS	LANGU	SYSTE	PROGR	4					
5	LISP	TECHN	PAGIN	ENVIR	4					
5	SPELL	CORRE	SYSTE	PROGR	4					
3	EXTEN	LANGU	4							
6	EFFIC	CONTE	FREE	PARSI	ALGOR	4				
6	PDEL	LANGU	PARTI	DIFFE	EQUAT	4				
4	FORMA	TRANS	INTER	4						
8	AMESP	HIGHE	LEVEL	DATA	PLOTT	SOFTW	SYSTE	4		
9	TECHN	GENER	OPTIM	FLOYD	EVANS	PRODU	PRECE	GRAMM	4	
4	AXIOM	APPRO	PROGR	4						
4	LANGU	TREAT	GRAPH	4						
7	GEDAN	SIMPL	TYPEL	LANGU	COMPL	REFER	4			
4	TRANS	MATRI	COMPI	4						
4	TRANS	WRITI	SYSTE	4						





8	GLOBAL	PARSE	CONTE	FREE	PHRAS	STRUC	GRAMM	4	
5	GENER	PURPO	GRAPH	LANGU	4				
5	CHAMP	CHARA	MANIP	PROCE	4				
4	ITRA	PROGR	LANGU	4					
5	AXIOM	BASIS	COMPU	PROGR	4				
6	PRACT	METHO	CONST	LR(K)	PROCE	4			
5	APARE	PARSE	REQUE	LANGU	4				
5	GENER	PROCE	PROGR	LANGU	4				
4	ARITH	EXPRE	TREE	4					
7	BLOCK	STRUC	INDIR	ADDRE	GARBA	COLLE	4		
4	MAD	DEFIN	FACIL	4					
5	ALGOL	BASED	ASSOC	LANGU	4				
6	LINEA	PRECE	FUNCT	PRECE	GRAMM	4			
6	ALGOR	CONST	BOUND	CONTE	PARSE	4			
5	ALGOL	CONST	PROCE	PARAM	4				
5	SYNTA	DIREC	DOCUM	PL360	4				
4	COMPL	CALCU	MATRI	4					
7	FINIT	ASSUM	INTEL	ISOLA	COMPU	SCIEN	4		
6	COMPL	MATRI	INVER	VERSU	REAL	5			
5	MULTI	PREC1	DIVIS	ALGOR	5				
5	CONDI	NUMBE	PEI	MATRI	5				
11	COMPU	POLYN	RESUL	BEZOU	DETER	VERSU	COLLI	REDUC	P.R.S
	ALGOR	5							
9	IMPRO	ROUND	OFF	RUNGE	KUTTA	COMPU	GILL	METHO	5
8	ALGOR	BOUND	GREAT	COMMO	DIVIS	N	INTEG	5	
7	LOGAR	ERROR	NEWTO	METHO	SQUAR	ROOT	5		
9	INTER	ARITH	DETER	EVALU	USE	TESTI	CHEBY	SYSTE	5
6	EXTRE	PORTA	RANDO	NUMBE	GENER	5			
9	ALGOR	SOLVI	SPECI	CLASS	TRIDI	SYSTE	LINEA	EQUAT	5
5	COMPU	JN(X)	NUMER	INTEG	5				
7	SIMPL	METHO	LINEA	PROGR	LU	DECOM	5		
8	FAST	FOURI	TRANS	ALGOR	REAL	VALUE	SERIE	5	
10	APPRO	SOLUT	INITI	BOUND	WAVE	EQUAT	PROBL	VALUE	TECHN 5
8	ONE	LINE	RANDO	NUMBE	GENER	USE	COMBI	5	
10	PRACT	ERROR	COEFF	INTEG	PERIO	ANALY	FUNCT	TRAPE	RULE 5
10	ADAPT	NEWTO	COTES	QUADR	ROUTI	EVALU	DEFIN	INTEG	PEAKE 5
8	ORIGI	SHIFT	QR	ALGOR	SYMME	TRIDI	MATRI	5	
4	INTEG	SQUAR	ROOT	5					
10	GOODM	LANCE	METHO	SOLUT	TWO	POINT	BOUND	VALUE	PROBL 5
10	OPTIM	START	APPRO	GENER	SQUAR	ROOT	SLOW	NO	DIVID 5
6	FORTR	TAUSW	PSEUD	NUMBE	GENER	5			
5	ALGOR	NONLI	MINIM	APPRO	5				
7	ERROR	IMPRO	ESTIM	ADAPT	TRAPE	INTEG	5		
5	CUBIC	SPLIN	UNIFO	MESH	5				
5	ACCUR	FLOAT	POINT	SUMMA	5				
13	STOPP	CRITE	NEWTO	RAPHS	METHO	IMPLI	MULTI	INTEG	ALGOR NONLI
	SYSTE	ORDIN	DIFFE						
15	EQUAT	5							
3	BINAR	SUMMA	5						
8	NUMER	PROPE	RITZ	TREFF	ALGOR	OPTIM	CONTR	5	



4COMPL	INTER	ARITH	5						
6AUTOM	INTEG	ORDIN	DIFFE	EQUAT	5				
5BEST	ONE	SIDED	APPRO	5					
9RAPID	COMPU	GENER	INTER	FORMU	MECHA	QUADR	RULE	5	
7MODIF	NORDS	METHO	OFF	STEP	POINT	5			
5ACCUR	FLOAT	POINT	SUMMA	5					
6RECUR	COMPU	DERIV	ERROR	PROPA	5				
9ROUGH	READY	ERROR	ESTIM	GAUSS	INTEG	ANALY	FUNCT	5	
8CHEBY	INTER	QUADR	FORMU	VERY	HIGH	DEGRE	5		
8SPLIN	FUNCT	METHO	NONLI	BOUND	VALUE	PROBL	5		
6RECUR	RELAT	DETER	PENTA	MATRI	5				
9GENER	TEXT	MATRI	SIGN	PATTE	PRESC	POSIT	SPECT	5	
6POLYN	SPLIN	APPRO	QUADR	PROGR	5				
7GENER	PSEUD	NUMBE	TWO'S	COMPL	MACHI	5			
4ACCEL	LP	ALGOR	5						
6ERROR	BOUND	PERIO	QUINT	SPLIN	5				
4ALGOR	FILON	QUADR	5						
3CHOIC	BASE	5							
4MINIM	LOGAR	ERROR	5						
5FORTR	RANDO	NUMBE	GENER	5					
3DOWNH	METHO	5							
5FAST	RANDO	NUMBE	GENER	5					
8PRACT	ERROR	COEFF	ESTIM	QUADR	ANALY	FUNCT	5		
4IN	OUT	CONVE	5						
6NUMER	INTEG	FORMU	FOURI	ANALY	5				
6QUASI	ESTIM	DIFFE	OPERA	EIGEN	5				
9STABL	NUMER	METHO	CHEBY	SOLUT	OVERD	SYSTE	EQUAT	5	
6METHO	CONVE	IMPRO	IMPRO	INTEG	5				
9DETER	INTER	POINT	TWO	PLANE	CURVE	DIFFE	EQUAT	5	
7GENER	POSIT	TEST	MATRI	KNOWN	SPECT	5			
8NUMER	SOLUT	THIN	PLATE	HEAT	TRANS	PROBL	5		
6CORRE	BEHAV	RANDO	NUMBE	GENER	5				



## APPENDIX 2

Distinct Keywords in the CACH Data Base.

PP. 119 - 121.



>	1	ACCEL	>	51	COMBI	>	101	ENVIR
>	2	ACCES	>	52	COMMA	>	102	EQUAT
>	3	ACCUR	>	53	COMMO	>	103	ERROR
>	4	ACTIV	>	54	COMMU	>	104	FSOPE
>	5	ADAPT	>	55	COMPA	>	105	ESTIM
>	6	ADDRE	>	56	COMPI	>	106	EVALU
>	7	ALGOL	>	57	COMPL	>	107	EVANS
>	8	ALGOR	>	58	COMPO	>	108	EVENT
>	9	ALLOC	>	59	COMPU	>	109	EXCLU
>	10	AMBIG	>	60	CONCU	>	110	EXECU
>	11	AMESP	>	61	CONDI	>	111	EXIST
>	12	ANALY	>	62	CONSO	>	112	EXPRE
>	13	ANOMA	>	63	CONST	>	113	EXTEN
>	14	APARE	>	64	CONTE	>	114	EXTRE
>	15	APPRO	>	65	CONTR	>	115	FACIL
>	16	ARITH	>	66	CONVE	>	116	FAST
>	17	ASSIG	>	67	CORRE	>	117	FILE
>	18	ASSOC	>	68	COTES	>	118	FILON
>	19	ASSUM	>	69	CRITE	>	119	FIND
>	20	ATTRI	>	70	CUBIC	>	120	FINIT
>	21	AUTOM	>	71	CURVE	>	121	FLOAT
>	22	AVERA	>	72	DATA	>	122	FLOYD
>	23	AXIOM	>	73	DEADL	>	123	FOLDI
>	24	BANK	>	74	DECOM	>	124	FORMA
>	25	BASE	>	75	DEFIN	>	125	FORMU
>	26	BASED	>	76	DEGRE	>	126	FORTR
>	27	BASIC	>	77	DEMAN	>	127	FOURI
>	28	BASIS	>	78	DENSE	>	128	FRAGM
>	29	BATCH	>	79	DERIV	>	129	FREE
>	30	BEHAV	>	80	DESIG	>	130	FULL
>	31	BEST	>	81	DETEC	>	131	FUNCT
>	32	BEZOU	>	82	DETER	>	132	GARBA
>	33	BINAR	>	83	DEVEL	>	133	GAUSS
>	34	BLISS	>	84	DIEFE	>	134	GEDAN
>	35	BLOCK	>	85	DIGIT	>	135	GENER
>	36	BOOLE	>	86	DIREC	>	136	GILL
>	37	BOUND	>	87	DISPL	>	137	GLOBA
>	38	CALCU	>	88	DISTP	>	138	GOODM
>	39	CANON	>	89	DIVID	>	139	GPL
>	40	CARD	>	90	DIVIS	>	140	GRAMM
>	41	CHAMP	>	91	DOCUM	>	141	GRAPH
>	42	CHARA	>	92	DOWNH	>	142	GREAT
>	43	CHEBY	>	93	DRIVE	>	143	HALF
>	44	CHOIC	>	94	DUPLE	>	144	HASHI
>	45	CLASS	>	95	DYNAM	>	145	HEAT
>	46	CODAS	>	96	EASY	>	146	HIEPA
>	47	CODIN	>	97	EFFIC	>	147	HIGH
>	48	COEFF	>	98	EIGEN	>	148	HIGHE
>	49	COLLE	>	99	ENGLI	>	149	IITRA
>	50	COLLI	>	100	ENTRY	>	150	IMPLE





>	151	IMPLI	>	201	MINIM	>	251	PL360
>	152	IMPRO	>	202	MODEL	>	252	POINT
>	153	IN	>	203	MODIF	>	253	POLIC
>	154	INDEX	>	204	MODUL	>	254	POLYN
>	155	INDIR	>	205	MONIT	>	255	PORTA
>	156	INDUC	>	206	MULTI	>	256	POSIT
>	157	INFOR	>	207	N	>	257	PRACT
>	158	INHER	>	208	NAME	>	258	PRECE
>	159	INITI	>	209	NATUR	>	259	PREC I
>	160	INSTP	>	210	NETWO	>	260	PREDI
>	161	INTEG	>	211	NEWTO	>	261	PRESO
>	162	INTEL	>	212	NO	>	262	PREVE
>	163	INTER	>	213	NONLI	>	263	PRINC
>	164	INVER	>	214	NORDS	>	264	PRIOR
>	165	ISOLA	>	215	NUCLE	>	265	PROBL
>	166	JN(X)	>	216	NUMBE	>	266	PROCE
>	167	KEY	>	217	NUMER	>	267	PRODU
>	168	KNOWN	>	218	OFF	>	268	PROGR
>	169	KUTTA	>	219	ON	>	269	PROOF
>	170	LANCE	>	220	ONE	>	270	PROPA
>	171	LANGU	>	221	OPERA	>	271	PROPE
>	172	LARGE	>	222	OPTIM	>	272	PSEUD
>	173	LAWYE	>	223	ORDER	>	273	PUNCH
>	174	LENGT	>	224	ORDIN	>	274	PURPO
>	175	LEVEL	>	225	ORGAN	>	275	QR
>	176	LEWIN	>	226	ORIGI	>	276	QUADR
>	177	LINE	>	227	OUT	>	277	QUASI
>	178	LINEA	>	228	OVERD	>	278	QUINT
>	179	LINK	>	229	OVERL	>	279	RANDO
>	180	LISP	>	230	P.R.S	>	280	RAPHS
>	181	LIST	>	231	PAGE	>	281	RAPID
>	182	LOADF	>	232	PAGI	>	282	READF
>	183	LOGAR	>	233	PAGIN	>	283	READY
>	184	LP	>	234	PARAL	>	284	REAL
>	185	LP(K)	>	235	PARAM	>	285	RECUR
>	186	LPLTR	>	236	PARSE	>	286	REDUC
>	187	LU	>	237	PARSI	>	287	REFER
>	188	MACHI	>	238	PARTI	>	288	RFLAT
>	189	MAD	>	239	PATTE	>	289	RELEV
>	190	MANAG	>	240	PDEL	>	290	RELIA
>	191	MANIP	>	241	PEAKE	>	291	REMOT
>	192	MANUA	>	242	PFEKA	>	292	RFOUE
>	193	MATRI	>	243	PEI	>	293	RESOU
>	194	MEANI	>	244	PENTA	>	294	RESUL
>	195	MEASU	>	245	PERFO	>	295	RETRI
>	196	MECHA	>	246	PERIO	>	296	RITZ
>	197	MEMOR	>	247	PHRAS	>	297	ROOT
>	198	MESH	>	248	PLANE	>	298	ROUGH
>	199	METHO	>	249	PLATE	>	299	ROUND
>	200	MICRO	>	250	PLOTT	>	300	ROUTI



>	301	RULE	>	351	TESTI
>	302	RUNGE	>	352	TEXT
>	303	SCHED	>	353	THEOR
>	304	SCIEN	>	354	THIN
>	305	SEARC	>	355	TIME
>	306	SEGME	>	356	TRADE
>	307	SEMAN	>	357	TRAFF
>	308	SERIE	>	358	TRANS
>	309	SET	>	359	TRAPE
>	310	SHARE	>	360	TREAT
>	311	SHARI	>	361	TREE
>	312	SHIFT	>	362	TREFF
>	313	SIDED	>	363	TRIDI
>	314	SIGN	>	364	TWO
>	315	SIMPL	>	365	TWO'S
>	316	SIMUL	>	366	TYPEL
>	317	SIZE	>	367	TYPEW
>	318	SLOW	>	368	UNIFO
>	319	SOFTW	>	369	UPDAT
>	320	SOLUT	>	370	USE
>	321	SOLVI	>	371	USER
>	322	SORTI	>	372	UTILI
>	323	SPACE	>	373	VALUE
>	324	SPECI	>	374	VARIA
>	325	SPECT	>	375	VERSU
>	326	SPEED	>	376	VERY
>	327	SPELL	>	377	WAVE
>	328	SPLIN	>	378	WORKI
>	329	SQUAR	>	379	WRITE
>	330	STABL	>	380	WRITI
>	331	STAND	>	381	68
>	332	START			
>	333	STATI			
>	334	STEP			
>	335	STOPP			
>	336	STORA			
>	337	STRAT			
>	338	STRUC			
>	339	SURFX			
>	340	SUMMA			
>	341	SYMMF			
>	342	SYNCH			
>	343	SYNTA			
>	344	SYNTH			
>	345	SYSTE			
>	346	TAUSW			
>	347	TECHN			
>	348	TELEP			
>	349	TELET			
>	350	TEST			



## APPENDIX 3

Classification Program.

PP. 123 - 129.



```

$$$SIGNON FKCC T=1M 9TP=2 'AUTOMATIC DOCUMENT CLASSIFICATION'
FKCHAN
$$RUN *FORTG

```

```

*****
*
*  AUTOMATIC DOCUMENT CLASSIFIER  *
*
*              PHASE 1              *
*
*****

```

# DECLARATION STATEMENTS

```

      INTEGER*2 KWORD(3,15),LIST(3,200),DOC(200)
      INTEGER*2 BLANK/' '/,NO13/13/,ZERO/' 0'/,SIX/' 6'/,
/KOUNT/0/,INDEX/0/
10  K=1

      READ A CARD FROM THE CARD FILE

20  READ(5,101,END=50)N,((KWORD(J,I),J=1,3),I=K,N)

      TEST FOR THE SPECIAL CASE

      IF(N.NE.NO13)GOTO 30

      TEST FOR NUMERIC CLASS NUMBER IN THE LAST LOGICAL
      RECORD

      IF(KWORD(1,N).GT.ZERO.AND.KWORD(1,N).LT.SIX.AND.
/KWORD(2,N).EQ.BLANK)GOTO 30

      MORE INFORMATION ON THE NEXT CARD

      K=N+1
      GOTO 20
30  KOUNT=KOUNT+1

      RECORD THE CORRECT CLASSIFICATION FOR EACH DOCUMENT ON
      DISK

      WRITE(3,104)KWORD(1,N)
      NN=N-1

      STORE UP THE KEYWORDS FOR EACH DOCUMENT ON DISK

      WRITE(4)NN,((KWORD(I,J),I=1,3),J=1,NN)

```





```

C
C   RECORD THE CORRESPONDING DOCUMENT NUMBER FOR EACH
C   KEYWORD
C
C   DO 40 I=1,NN
C   INDEX=INDEX+1
C   DOC(INDEX)=KOUNT
C
C   RECORD THE KEYWORDS FOR EACH DOCUMENT
C
C   DO 40 J=1,3
C   40 LIST(J,INDEX)=KWORD(J,I)
C   GOTO 10
C
C   RECORD ALL THE KEYWORDS AND THEIR CORRESPONDING
C   DOCUMENT NUMBERS ON DISK
C
C   50 DO 60 I=1,INDEX
C   60 WRITE(2,102)(LIST(J,I),J=1,3),DOC(I)
C
C   RECORD THE COUNTS FOR THE NUMBER OF KEYWORDS AND THE
C   NUMBER OF DOCUMENTS ON DISK
C
C   WRITE(8,103)INDEX,KOUNT
C   STOP
C
C   FORMAT STATEMENTS
C
C   101 FORMAT(I2,13(3A2))
C   102 FORMAT(3A2,I2)
C   103 FORMAT(I3,I2)
C   104 FORMAT(A2)
C   END
C   $$ENDFILE

```



```
$$CREATE -IN TYPE=SEQ  
$$CREATE -CLASS TYPE=SEQ  
$$CREATE -DOCU TYPE=SEQ  
$$CREATE -NUMBER TYPE=SEQ  
$$RUN -LOAD# 2=-IN 3=-CLASS 4=-DOCU 8=-NUMBER
```

DOCUMENTS TO BE CLASSIFIED

```
$$ENDFILE  
$$CREATE -OUT TYPE=SEQ  
$$RUN *SORT  
SORT=CH;A;1;5;CH;A;7;2  
INPUT=-IN;U;20;20  
OUTPUT=-OUT;U;20;20  
MNR=200  
$$ENDFILE
```



\$\$RUN \*FORTG

C  
C  
C  
C  
C  
C  
C  
C  
C  
C  
C  
C  
C

```
*****  
*                                     *  
* AUTOMATIC DOCUMENT CLASSIFIER    *  
*                                     *  
*                               PHASE 2                                *  
*                                     *  
*****
```

## DECLARATION STATEMENTS

```

INTEGER*2 CLASS(25),DKWORD(3,381),LIST(3,200),DOC(200)
/,DOCU(3,15),IDENT(2,25)
REAL*4 FMAT(20,25),CDB(20,381),INV(20,20),MEAN(20),
/DUMMY(20),B(5),FUNCT(5)
DATA NUM/20/,JSTEP/1/,MAG/381/,NO/25/
SMALL=-0.5*10.0**70

```

C  
C  
C

INITIALIZE FMAT TO ALL ZEROS

```
DO 150 I=1,NO
DO 150 J=1,NUM
150 FMAT(J,I)=0.0
```

C  
C  
C

READ THE COMPRESSED DATA BASE FROM TAPE

```

REWIND 1
DO 10 I=1,NUM
10 READ(1)(CDB(I,J),J=1,MAG)

```

C  
C  
C

## READ THE DISTINCT KEYWORDS FROM DISK

```
DO 20 I=1,MAG
20 READ(2,101)(DKWORD(J,I),J=1,3)
```

C  
C  
C  
C

READ THE COUNTS FOR THE NUMBER OF KEYWORDS AND THE  
NUMBER OF DOCUMENTS FROM DISK

READ(7,102)INDEX,KOUNT

C  
C  
C  
C

READ THE CORRECT CLASSIFICATION FOR EACH DOCUMENT FROM DISK

```
DO 19 J=1,KOUNT
19 READ(3,103)CLASS(J)
```

C  
C

READ A KEYWORD AND ITS CORRESPONDING DOCUMENT NUMBER



```

C      FROM DISK
C
      DO 30 I=1,INDEX
      READ(4,104)(LIST(K,I),K=1,3),DOC(I)
70 DO 40 N=1,3
C
C      COMPARE THE KEYWORD WITH A KEYWORD IN THE DISTINCT
C      KEYWORD LIST
C
      IF(DKWORD(N,JSTEP).LT.LIST(N,I))GOTO 50
C
C      TEST WHETHER THAT KEYWORD IS ABSENT IN THE DISTINCT
C      KEYWORD LIST
C
      IF(DKWORD(N,JSTEP).GT.LIST(N,I))GOTO 30
40 CONTINUE
C
C      KEYWORD IS PRESENT IN THE DISTINCT KEYWORD LIST, THE
C      CONTRIBUTIONS OF THAT KEYWORD IS ADDED TO THE FEATURE
C      VECTOR
C
      MS=DOC(I)
      DO 60 LINK=1,NUM
60 FMAT(LINK,MS)=FMAT(LINK,MS)+CDB(LINK,JSTEP)
      GOTO 30
C
C      COMPARE THAT KEYWORD WITH THE NEXT KEYWORD IN THE
C      DISTINCT KEYWORD LIST
C
50 JSTEP=JSTEP+1
      GOTO 70
30 CONTINUE
C
C      READ THE CONSTANTS FOR THE FIVE CLASSES
C
      READ(5,93)(B(IX),IX=1,5)
C
C      READ THE STATISTICS FOR CLASSIFICATION FROM TAPE
C
      DO 90 I=1,KOUNT
      REWIND 8
      DO 80 J=1,5
C
C      READ THE MEAN FEATURE VECTOR FOR A PARTICULAR CLASS
C
      READ(8)(MEAN(K),K=1,NUM)
C
C      READ THE INVERSE OF THE COVARIANCE MATRIX FOR A
C      PARTICULAR CLASS
C

```





```

      DO 207 K=1,NUM
207  READ(8)(INV(N,K),N=1,NUM)
C
C      PERFORM THE NECESSARY CALCULATIONS FOR THE
C      DISCRIMINANT FUNCTION
C
      DO 209 L=1,NUM
209  MEAN(L)=FMAT(L,1)-MEAN(L)
      DO 210 M=1,NUM
      DUMMY(M)=0.0
      DO 210 N=1,NUM
210  DUMMY(M)=DUMMY(M)+MEAN(N)*INV(N,M)
      RESULT=0.0
      DO 211 L=1,NUM
211  RESULT=RESULT+MEAN(L)*DUMMY(L)
C
C      OBTAIN THE NUMERIC VALUE FOR THE DISCRIMINANT FUNCTION
C      OF A PARTICULAR CLASS
C
      80  FUNCT(J)=B(J)-0.5*RESULT
C
C      ASSIGN A CLASS NUMBER TO THE DOCUMENT BASED ON THE
C      VALUES OF ITS DISCRIMINANT FUNCTIONS
C
      DO 90 N=1,2
      BIG=FUNCT(1)
      ID=1
C
C      FIND THE LARGEST VALUE OF THE DISCRIMINANT FUNCTIONS
C
      DO 515 K=2,5
      IF(FUNCT(K).LT.BIG)GOTO 515
      BIG=FUNCT(K)
      ID=K
515  CONTINUE
C
C      FIND THE SECOND LARGEST VALUE OF THE DISCRIMINANT
C      FUNCTIONS
C
      IDENT(N,1)=ID
      FUNCT(ID)=SMALL
      90  CONTINUE
C
C      PRINT THE HEADINGS FOR THE OUTPUT
C
      WRITE(6,105)
      WRITE(6,106)
C
C      PRINT OUT THE CLASSIFICATION FOR EACH DOCUMENT
C

```



```

DO 700 KK=1,KOUNT
C
C RETRIEVE THE KEYWORDS FOR EACH DOCUMENT FROM DISK
C
  READ(9)NN,((DOCU(I,J),I=1,3),J=1,NN)
  WRITE(6,107)((DOCU(I,J),I=1,3),J=1,NN)
  WRITE(6,108)CLASS(KK),(IDENT(I,KK),I=1,2)
700 CONTINUE
  STOP
C
C FORMAT STATEMENTS
C
101 FORMAT(3A2)
102 FORMAT(I3,I2)
103 FORMAT(A2)
104 FORMAT(3A2,I2)
105 FORMAT('1',//19X,'KEYWORDS FROM EACH DOCUMENT',48X,
  /'CORRECT CLASS',6X,'ASSIGNED CLASS'/)
106 FORMAT(' ',110X,'1ST RANK',4X,'2ND RANK'/)
  93 FORMAT(5E14.4)
107 FORMAT('0',2X,15(3A2))
108 FORMAT('+',T101,A2,T115,I2,T127,I2)
  END
$$ENDFILE
$$RUN *MOUNT
0007 ON 9TP *TAPE3* VOL=FC0007 RING=OUT LRECL=255
BLKSIZE=5100 FMT=FB 'NEXT 0030'
0030 ON 9TP *TAPE4* VOL=FC0030 RING=OUT LRECL=255
BLKSIZE=5100 FMT=FB
$$ENDFILE
$$RUN -LOAD# 1=*TAPE4* 2=KWORD 3=-CLASS 4=-OUT 8=*TAPE3*
9=-DOCU 7=-NUMBER
      0.4174E 02      0.5139E 02      0.4470E 02      0.3231E 02
      0.2037E 02
$$ENDFILE
$$SIGNOFF

```



## APPENDIX 4

Experimental Classification Results.

PP. 131 - 135.



## KEYWORDS FROM EACH DOCUMENT

	CORRECT CLASS	ASSIGNED CLASS	
		1ST RANK	2ND RANK
RELIA FULL	1	1	4
TIME SHARI BATCH PROCE COMPA VALUE PROBL SOLVI	1	1	5
STORA UTILI MEMOR HIERA ASSIG PERFO PASHI ALGCR	1	1	3
ANALY TIME SHARI TECHN	1	1	4
MODUL CCMPU SHARI SYSTE	1	1	4
VARIA LENGT TREE STRUC MINIM AVERA SEARC TIME	2	2	4
MULTI ATTRI RETRI CCMBI INDEX	2	2	3
CODAS DATA DISPL SYSTE	2	2	5
DATA BASE DEADL	2	2	3
AVERA BINAP SEARC LENGT DENSE ORDER LIST	2	2	4
PREVE SYSTE DEADL	3	3	4
USER PRGR MEASU TIME SHARE ENVIR	3	3	1
MEANI NAME PRGR SYSTE	3	3	4
POLIC DRIVE SCHED TIME SHARI SYSTE	3	3	4
INTER COMMA GENER FACIL	3	3	4
ALGOR CONST BOUND CONTE PARSE	4	4	5
ALGOL CONST PROCE PARAM	4	3	4
SYNTA DIREC DOCUM PL360	4	4	5
COMPL CALCU MATRI	4	4	5
FINIT ASSUM INTEL ISOLA COMPU SCIEN	4	4	5
BEST ONE SIDED APPROG	4	4	5
RAPID COMPU GNER INTER FORMU MECHA QUADR RULE	5	5	4
MODIF NURDS METHU OFF STEP POINT	5	5	4
ACCUR FLOAT POINT SUMMA	5	5	4
RECUR COMPU DERIV ERROR PROPA	5	4	5





KEYWORDS FROM EACH DOCUMENT		CORRECT CLASS		ASSIGNED CLASS	
				1ST RANK	2ND RANK
SIMUL NETWO PARAL PROCE EVENT		1		1	5
INTER GRAPH DISPL MONIT BATCH PRCE ENVIR REMCT ENTRY		1		3	5
OPTIM DESIG COMPU GRAPH SYSTE		1		1	5
RELIA FULL DUPLR FILE TRANS HALF TELEP LINE		1		1	4
DYNAM MICRO PROCE ORGAN PRGCR		1		1	4
OPTIM TREE SIRUC		2		2	4
SYNCH PARAL ACCES DATA BASE		2		2	4
LEWIN ORDER RETRI THEOR ASSOC MEMOR		2		2	4
COMPU LAWYE		2		2	3
PEEKA PUNCH CARD NATUR LANGU SEARC		2		2	4
PROCE MANAG RESOU SHARI MULTI ACCES SYSTE ESOP		3		3	4
INTER BASED ORGAN MANAG INEOR SYSTE		3		3	4
ORGAN MATRI OPERA PAGE MEMOR SYSTE		3		3	5
PSEUD		3		3	4
DYNAM SPACE SHARI COMPU SYSTE		3		3	4
IIITRA PROGR LANGU		4		4	5
AXICM BASIS COMPU PROGR		4		4	3
PRACT METHO CONST IRIKJ PROCE		4		4	3
APARE PARSE REQUE LANGU		4		4	5
GENER PRCE PROGR LANGU		4		1	4
FORIR TAUSW PSEUD NUMBE GENER		5		5	4
ALGOR NONLI MINIM APPRO		5		4	5
ERROR IMPRO ESTIM ADAPT TRAPE INTEG		5		5	4
CUBIC SPLIN UNIFO MESH		5		4	3
ACCUR FLOAT POINT SUMMA		5		5	4



KEYWORDS FROM EACH DOCUMENT		CORRECT CLASS		ASSIGNED CLASS	
				1ST RANK	2ND RANK
STORA PRACH PROGR SEGMP		1		1	3
AUTOM FOLDI PROGR EFFIC MANUA		1		1	4
DEGRE MULTI PAGIN ON DEMAN SYSTE		1		5	3
PROGR PARAI PRCCE		1		1	4
ESTIM DISTR PANDO VARIA COMPU COMMU TRAFF MODEL		1		1	5
CANON STRUC ATTRI BASED FILE ORGAN		2		3	2
KEY ADDRE TRANS TECHN LARGE EXIST FORMA FILE		2		2	4
RETRI UPDAT SPEED TRADE COMBI INDEX		2		2	3
RANDO BINAR SEARC TECHN		2		2	4
RETRI TIME DIREC ACCES INVER FILE		2		4	5
CONVE ACCES MACHI		3		3	4
ANOMA PAGIN		3		3	4
SORTI PAGIN ENVIR		3		3	4
INSTR MULTI		3		3	4
WORKI SET MODEL PRGGR BEHAV		3		3	4
ARITH EXPRE TREE		4		4	5
BLOCK STRUC INDIR ADDRE GARBA CCLLE		4		4	5
MAD DEFIN FACIL		4		5	4
ALGOL BASED ASSOC LANGU		4		4	5
LINEA PRECE FUNCT PRECE GRAMM		4		4	3
STOPP CRITE NEWTC RAPHs METHO IMELI MULTI INTEG ALGOR NONLI SYSTE ORDIN DIFFE EQUAT		5		5	4
BINAR SUMMA		5		5	4
NUMER PROPE RITZ TREFF ALGOR OPTIM CONTR		5		5	3
COMPL INTER ARITH		5		1	3
AUTOM INTEG ORDIN DIFFE EQUAT		5		5	4



KEYWORDS FROM EACH DOCUMENT		CORRECT CLASS		ASSIGNED CLASS	
		1ST RANK	2ND RANK	1ST RANK	2ND RANK
EXCLU SIMUL ACTIV DIGIT NETWO		1	1	5	
DESIG DISPL PROCE		1	1	5	
ANALY-BCCLE-PROGR-MODEL-TIME SHARI-PAGIN-ENVIR		1	3	4	
SUBEX ORDER EXECU ARITH EXPRE		1	1	2	
INTER DRIVE PROGR		1	3	4	
INFOR-REIRI-TELET		2	3	4	
EASY ENGLI LANGU INFOR RETRI REMOT TYPE# CONSO		2	2	5	
RELEV ESTIM IMPRO		2	1	2	
FORMA-SYSIE-INFOR-REIRI-FILE		2	3	4	
RELAT MODEL DATA LARGE SHARE DATA BANK		2	2	4	
ANOMA SPACE TIME CHARA PROGR PAGIN MACHI		3	3	4	
PREVE-SYSIE-DEADL		3	3	4	
MULTI MACHI CODIN COMPU ORGAN		3	3	4	
LOADE STAND OVERL PROGR		3	3	4	
PERED-MONIT-TIME SHARI-SYSIE		3	3	5	
TRANS MATRI COMPI		4	4	5	
TRANS WRITI SYSTE		4	4	3	
GLOBAL-PARSE-CONTE-FREE PHRAS-SIRUC-GRAMM		4	4	1	
GENER PURPO GRAPH LANGU		4	1	4	
CHAMP CHARA MANIP PROCE		4	1	4	
ADAPI-NEMIO-COTES-QUADR-ROUTI-EVALU-DEFIN-INTEG-PEAKE		5	5	4	
ORIGI SHIFT QR ALGOR SYMME TRIDI MATRI		5	5	4	
INTEG SQUAR ROOT		5	5	4	
GOODM-LANCE-METHO-SOLUT-IWO POINT-BCUNC-VALUE-PROHL		5	5	4	
OPTIM START APPRO GENER SQUAR ROOT SLOW NO DIVID		5	5	4	



## KEYWORDS FROM EACH DOCUMENT

	CORRECT CLASS	ASSIGNED CLASS	
		1ST RANK	2ND RANK
DYNAM MICRO PROCE ORGAN PROGR	1	1	4
SYSTE INTER COMMU RESCU SHARI COMPU NETWO	1	1	5
ENVIR RESEA MICRO EMULA	1	3	4
OPTIM PERFO TIME CHARI SYSTE SIMUL	1	1	5
DEMAN PAGIN UTILI WORKI SET MANIA	1	3	4
FILE ORGAN CONSE RETRI PROPE	2	3	5
DESIG VENUS OPERA SYSTE	3	3	4
HARDW ARCHI IMPL EPCYE RING	3	3	1
STORA PARTI MATHE MODEL LOCAL	3	3	4
TENEX PAGED TIME SHARI SYSTE	3	3	4
MUX ONLI COMPU	3	1	3
MULTI VIRTU MEMOR CONCCE DESIG	3	3	4
PROPE WORKI SET MODEL	3	3	5
OPERA SYSTE CONCCE SUPER COMPU	3	3	4
COMPA ANALY DISK SCHED POLIC	3	4	5
BOOLE MATRI METHO COMPU LINEA PRECF FUNCT	4	4	5
MODEL TYPE CHECK APELLI ALGOL 60	4	3	4
DERIV SEMAN PROGR LANGU CONST	4	4	5
GENER PARSE AFFIX GRAMM	4	4	5
BLOCK DATAT SNOBO	4	4	5
CONVE LIMIT ENTRY DECIS TABLE OPTIM NEAR FLOWC ALGOR	5	5	4
SORTI NATUR SELEC	5	2	4
CONVE DECIS TABLE RULE MASK METHO WITHO	5	4	5
SORTI PROBL COMPL	5	1	5
TECHN SOFTW MODUL SPECI	5	4	5

















**B30045**